Structuring Free-Form Building Envelopes

by

Daniel Gebreiter

Supervised by

Dr. Paul Shepherd
Prof. Paul Richens

A thesis submitted for the degree of Master of Philosophy
University of Bath
Department of Architecture and Civil Engineering
September 2012

Acknowledgements

I would like to express sincere gratitude to Dr. Paul Shepherd and Prof. Paul Richens for their excellent supervision, guidance and patience over the course of this year. I'd also like to acknowledge Schlaich Bergermann and Partners for funding this project. In particular, I owe thanks to Prof. Roman Kemmler for his curiosity and asking the right questions, Sven Plieninger and Uwe Burkhardt for producing new and ever interesting case studies, as well as Stefan Justiz for his continuous support. And to Zuzana of course.

Abstract

In contemporary architectural design, the definition of envelope geometry often precedes consideration of its material build-up. The geometries created in such a top-down manner require subsequent rationalisation for construction.

This thesis addresses methods for the aesthetic and efficient structuring of arbitrary surfaces while respecting distinct architectural considerations. Digital design tools have been developed to approximate surfaces using isotropic triangular or quadrangular elements. Various methods for the creation of simplified base complexes are shown. Such complexes expose a simplified topological build-up of surfaces and are used to create global parameterisations of surfaces of arbitrary genus. For quadrangular discretisation, the restructuring can be aligned to a user-defined vector field on the surface. The generated geometries are then optimised with regard to architectural considerations by means of dynamic relaxation.

Particular attention is given to the compatibility of the methods with existing design workflows. Suitability and performance of the tools are assessed when used for the design of a real glass roof designed with Schlaich Bergermann and Partner in Luwan, China.

Figures

Contents

1 Introduction

The discretisation of surfaces plays a crucial role in the aesthetics, as well as the manufacture of architecture. Traditional envelope design subjects geometries to pre-conceived rationales and material constraints when built from modular elements. Supported by advances in construction, state-of-the-art computational design tools permit the inversion of this bottom-up approach. Methods have been proposed to efficiently generate topological complexes over arbitrary surfaces which at once approximate the desired geometry very closely, and meet stringent architectural requirements regarding aesthetics and constructability.

This thesis addresses the efficient structuring and panelling of arbitrary surfaces considering the above top-down approach in particular. For illustration, Figure 1 shows the surface of a glass roof discretised from triangular (middle) and quadrangular elements (right).

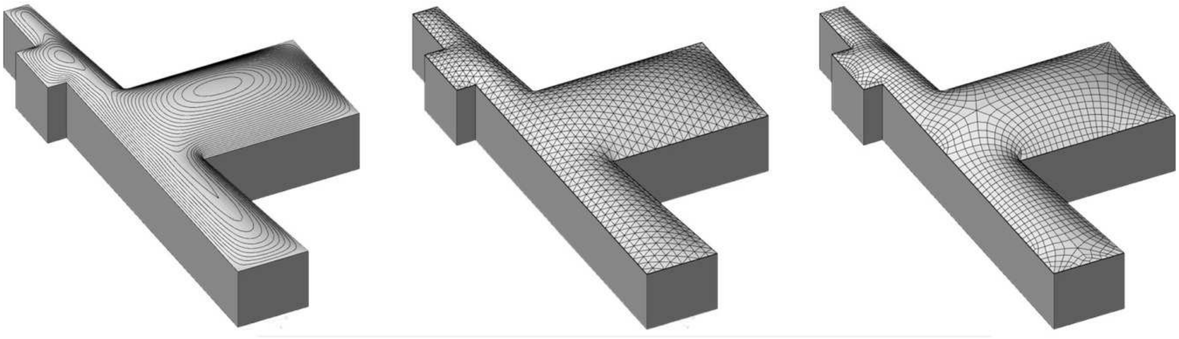

Figure 1 Restructured glass roof surface. Left: The input surface, shown with contours. Middle: The geometry is approximated with triangular elements. Right: Quadrangular elements following principal curvature lines. Restructuring was done using the software tools outlined in this thesis. Input model (left) courtesy of Schlaich Bergermann and Partners.

1.1 Structuring architectural surfaces

Building envelopes are, with notable exceptions of true concrete shell and fabric structures commonly made up from modular elements. Cost-effectiveness is often achieved by the repetition of equal or similar elements. As a result, geometries of building envelopes have tended to follow restrictive geometric principles.

Enabled by intuitive digital modelling and the digitally enhanced rapid prototyping tools that have emerged over recent years, architects have seemingly been liberated from such constraints on their designs. This trend has largely detached the form-finding process from material systems, culminating in “anything-goes”, or rather “anything-must”, digital exuberance (Colletti, 2010). Material rationales are often introduced only in the later stages of the design development, penalising deviation from the standard mode of production.

Advances in manufacture such as CNC milling, laser cutting, water-jetting and cold- and hot-forming of glass or other materials have relaxed the requirement for a repetition of identical elements in construction. Instead, it is the more flexible and often automated manufacturing processes which are repeated, tailored to customise each individual building block (Carpo, 2011). In the domain of building envelopes, and particularly glass roofs,

steel node design has evolved rapidly, with many solutions combining geometric freedom with visual elegance (Glymph and Shelden, 2002) and even good structural performance (Hwang and Knippers, 2010).
As such, limits as to size and similarity of elements are still imposed by the processes involved. But rather than being a drawback, these limits coincide with a tendency to deem intricate self-similar or quasi-regular structures aesthetic.
Several other aspects of the applied construction process entail further stipulations of material and therefore the method of discretisation of surfaces used. These include the transportability of elements, local availability of material and labour, as well as cost of manufacture and assembly.
Many of these factors cannot be accurately quantified at the outset and might change during the design process, only one example being fluctuating steel prices. The increased flexibility gained by detaching the boundary representation from material system at early design stages therefore constitutes a rather pragmatic justification for the use of top-down approach in architectural design.

In the top-down methodology, surfaces defined in the initial envelope design stage must later be endowed with a discrete structure reflecting the material system used.
There are two distinct but related sub-classes to this problem in the architectural setting: the generation of suitable segmentations of arbitrary surfaces on one hand, and the cladding of said surfaces on the other. These tasks will be referred to as structuring and panelling respectively. The latter involves compliance with additional, material-dependent requirements with regards to planarity of elements or build-up of supporting structure.

By far the predominant method employed in practice is the manual reconstruction and structuring of the input surface. Often the desired shape is post-rationalised via its approximation using segments of simpler or singly-curved geometries, in effect replicating bottom-up strategies. This can be cumbersome and difficult, in particular when surfaces are discontinuous or of high topological complexity or curvature. It is very time-consuming for large-scale projects, and subject to tedious reworking as designs evolve.

It will be shown that restructuring can be achieved on the basis of a global parameterisation of the surface. Since such parameterisation might not be possible for all surfaces, they are decomposed into sub-domains which are individually restructured. To decompose input surfaces, base complexes are applied, revealing a simplified topological build-up. The individual sub-domains can be parameterised and structured individually, and a global restructuring achieved by their reassembly on the input surface. The correct choice of base complex is crucial to successful restructuring, placing methods to their creation at the centre of this thesis.
Figure 2 an shows input surface, a discrete representation of a base complex, as well as the final restructuring result for a simple geometric figure.

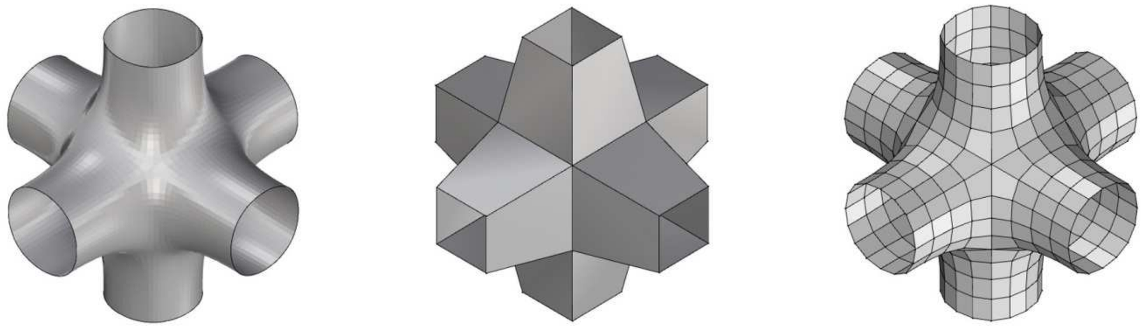

Figure 2 Restructuring from base complex. Left: Input surface, Middle: Simplified, quadrangular base complex. Right: The surface restructured from quadrilaterals only.

1.1.1 Goals and Contribution

In this thesis, a review of relevant literature is followed by a detailed description of the implementation of digital design tools for the construction of triangular and quadrangular base complexes for the global parameterisation of arbitrary surfaces. This parameterisation is used to rebuild an approximation of the input geometry from regular and isotropic (near equilateral) triangular or (near rectangular) quadrangular elements.

Rather than create a single tool, a modular framework of geometry processing tools is proposed to better address the unpredictable nature of architectural design tasks and constraints. The software can be used to generate geometries, visualise information to aid the structuring of surfaces, find topological base complexes or to optimise existing geometry. The tools are purposely embedded in the widely used parametric modelling package Grasshopper, a plug-in to Rhinoceros 3D (McNeel, 2011). Grasshopper was chosen because of its prevalence in the design community and its open programming interface.

Triangle meshes are used to represent geometry and perform computations. Meshes are discrete, piecewise linear approximations of input geometry but continuous surface properties can be reconstructed from the sampled data (see Appendix A). For the work presented in this thesis, techniques from digital mesh processing permitted the creation of a unified conduit for the versatile formulation of a whole range of operations referencing the same computational structure based on triangle meshes.
Adopting the vocabulary of meshes, the principal goal is to *remesh*, i.e. to restructure, input meshes according to architectural considerations. An example of a surface restructured from a triangular and a quadrangular base complex is illustrated in Figure 1.

Several techniques for remeshing using triangular and quadrangular base complexes are presented. For triangle meshes, base complexes are generated by reducing the vertex count of the input mesh.
It is shown how vector fields defined on the surface can serve to guide quadrangular remeshing. Techniques for the generation and visualisation of such fields are presented. Base complexes are then be generated manually, by tracing streamlines of the vector fields, or by using periodic global parameterisation techniques. The latter segments surfaces along the iso-parameter curves of periodic functions defined on the input geometry.

Multiple papers have addressed the issue of remeshing from a point of view of technical advances (Botsch et al., 2010, Hormann et al., 2008). In this thesis, the concern lies equally with the suitability of each method to interact within established building design workflows.

The tools have been employed for the design of a glass canopy in an actual building project in Shanghai, China, designed with Schlaich Bergermann and Partners. The suitability and performance of various triangular and quadrangular grid layouts for the free-form geometry are evaluated. Furthermore, reliability, advantages and shortcomings of the remeshing tools when used within a real project workflow are documented and assessed.

1.1.2 Overview

In section 1.2, literature on the subject of free-form construction and panelling of preconceived architectural surfaces is reviewed. The framework of the software proposed is introduced in chapter 0. In chapter 0, concepts from computer graphics and mathematics relevant to the construction of triangular base complexes are discussed. Using these base complexes, meshes can be globally parameterised and meshes extracted using the methods illustrated in chapter 4.

Three different automated quad-meshing techniques were implemented as part of this thesis. Spectral mesh processing is appraised for the task of remeshing without orientation control. The Morse-Smale Complex used to extract full quad meshes is described in chapter 0.

For two additional methods, vector fields are then used to guide the orientation of quad meshes. Literature on vector field processing on discrete surfaces is consequently reviewed in chapter 0, followed by an illustration of our tools for their creation.

Vector field streamline extraction for the creation of quad-dominant meshes is delineated in section 8.2. Finally, a wave-based parameterisation algorithm with vector field alignment is used in section 8.3 to decompose a surface into quadrangular domains.

An alternative method for meshing from a manually created base complex using subdivision with specially defined boundary constraints is proposed in chapter 0.

The remeshing tools are complemented by several tools for the subsequent refinement and optimisation of triangle and quad meshes. In particular, the multi-objective dynamic relaxation suite described in chapter 0 can be used for intuitive smoothing, fairing and planarisation of mesh elements.

In chapter 0, the performance and reliance of the remeshing and optimisation methods implemented is appraised when embedded into the workflow of an actual building project.

1.2 Previous work

In this section, previous work relating specifically to architectural applications of structuring and discretisation of surfaces is reviewed. In the later sections of this thesis, techniques are transferred from other fields, and from computer graphics in particular. Relevant literature is presented at the start of each chapter. Specifically, recent work on quadrangular remeshing is reviewed in section 5.1.2. The required vector field design on discrete surfaces is treated in section 8.1.2.

The cladding of surfaces using triangular panels has become commonplace for free-form geometries, and for glass roofs in particular. Quadrangular cladding requires the distinction between bottom-up and top-down approach to be made more austerely. This has been addressed in literature and buildings alike.

In architecture, the proof-of-concept often comes from successfully prototyping a new technology on an actual building. Free-form or lightweight glass roofs have become a popular choice for ambitious clients. Some approaches to their structuring are reviewed below.

1.2.1 Triangular discretisations

Triangular grids have become the norm for free-form claddings, steel and glass roofs in particular and innovation is frequent in this expansive field.

The grid of the British Museum Great Court roof by Foster and Partners was generated by draping a preconceived topological set-up over an analytically defined surface using dynamic relaxation (Williams, 2001). Similar draping can be achieved with the dynamic relaxation (Barnes et al., 1977) software outlined in chapter 0 of this thesis.

For the Trade Fair in Milan by Massimilano Fuksas architects, Schlaich Bergermann and Partners use iterative relaxation techniques to generate a regular mesh over a free-form NURBS-surface (Schober et al, 2004). Knippers and Partners Engineers restructure a free-form surface developed with Massimilano Fuksas architects using a manually created simplified base complex and subdivision to attain control over number and placement of singularities for the geometrically very complex MyZeil project in Frankfurt (Knippers and Helbig, 2010).

1.2.2 Bottom-up quadrangulation

Engineering consultancy Schlaich Bergermann and Partners have published comprehensive literature on the geometric construction and realisation of building fabrics based on the construction of translational, rotational and scaled surfaces, or combinations of these geometric operations. Such surfaces, if regularly sampled in the direction of their principal curvatures, yield isotropic planar faces (Schlaich and Schober, 2002, Schober, 1998), but the constraints of this bottom-up approach are obviously prohibitive when strong boundary conditions must be met, for example when interfacing with adjacent buildings.

Buildings

The principles of translational surfaces have been applied to many glass buildings. The earliest example is the Flusspferdehaus at Berlin Zoo by Schlaich Bergermann and Partners, whose blended doubly-curved cupolas are clad in planar quadrilateral glass panes. Continuous steel cables brace the structure diagonally to permit shell action (Schober, 1998).

The two conservatories of the Gardens by the Bay project in Singapore, by Wilkinson Eyre Architects constitute a recent example of a rotational surface discretised into planar quadrilaterals. The larger glass house is a segment of a purely rotational surface, whilst the smaller was scaled in one dimension during the design process. Scaling the roof in only one dimension retained its translational properties (Schlaich and Schober, 2002). Figure 3 shows a glass roof design based on a translational surface nearing completion in Paunsdorf, Germany.

1.2.3 Top-down quadrangulation

Whilst free-form architecture has gained in popularity, specific problems encountered in its top-down design have only recently been restated more formally in systematic mathematical terms (Pottmann, 2008, Wallner and Pottmann, 2011). A concise overview of methods linking geometry to numerical optimisation, including efficient planar quad mesh design, multi-layer construction and circle packing on architectural surfaces have been developed by Wallner and Pottmann (2011). Eigensatz et al. (2010b) propose an algorithm to find repetitive moulds for panels approximating an input mesh. They illustrate their algorithm, which also optimises the curve network to fit the restrictive pool of moulds, using several real-world examples (Eigensatz et al., 2010a).

Pottmann et al. (2008) develop algorithms based on the optimisation of b-splines to clad free-form surfaces with single curved elements based on a geometric description of developable strip models. Similarly, Flöry and Pottmann (2010) develop parametric models for the generation of curvature continuous surfaces generated from ruled surface strips only. Algorithmic methods for panelling using near-rectangular panels are illustrated by Wallner and Schiftner (2010) who further show shapes achievable via bending only.

Zadravec and Schiftner (2010) use level sets to approximate an input surface using regularly spaced quad-dominant meshes, and proceed to optimise these with a view toward planarity. They approach the problem by looking at conjugate (i.e. two orthogonal everywhere) vector fields and their singularities. A similar method has been implemented as part of this project and is described in section 8.3.

More recently, Liu et al. (2011) employ rotationally symmetric (RoSy) vector fields (see section 8.1) to generate planar quad-meshes for several architectural case studies.

Helmut Pottmann and French engineering consultancy RFR caused weighty debates within the design community when they applied for patents on the panelling of free-form surfaces with planar elements (Brell-Cokcan, 2007).

Buildings

The first quadrangular discretisation of freeform surfaces was proposed by geometer Gaspard Monge around 1750 (Sakarovitch, 2009) who illustrated the cellular decomposition of an ellipsoid by following principal curvature lines. Interested in the application of geometry to the real world, Monge proposed a design for the Palace of Legislation for the government emerging after the French Revolution. It featured a roof in the shape of the discretised ellipsoid now named after him, with chandeliers hanging from the ceiling at the locations of curvature field singularities (Sakarovitch, 2009).

1.2.4 Optimisation

Geometries are frequently optimised once the discrete topological build-up is determined. Objectives include the simplification of nodes or planarity of panels.

Bo et al. (2011) reduce node complexity of glass roofs by generalising meshes using arc segments rather than straight edges only. They further optimise such meshes to using repetitive radii and illustrate volumetric versions.

The mechanical properties of the grid layout are taken into account and improved by Schiftner and Balzer (2010) by the alignment of grid members with a vector field generated from simple stress calculations.
Liu et al. (2006) introduce conical meshes, which exhibit constant face offsets and torsion-free nodes, making them particularly suitable for multi-layer construction (Pottmann et al., 2007). As the layers of a "conical" envelope are at constant distance from another, repeating support details can be used.

Buildings

A notable example of post-rationalisation during the design process is the grid-shell hovering above the Yas Marina Hotel by Asymptote. It exhibits torsion-free steel nodes, i.e. the webs of all rectangular bars meeting at a node are aligned to a common axis. The grid, engineered by Schlaich Bergermann, Evolute and Waagner Biro, was developed using local subdivision techniques.
Non-planar quadrilateral elements make up the façade of the Zaha Hadids Nordpark Cable Railway stations in Innsbruck. Evolute GmbH helped develop a rational decomposition of the highly doubly-curved surfaces into quadrangular sub-domains. Moulds for the individual doubly-curved glass sheets were CNC-milled and the glass subsequently thermoformed until the desired tolerance was achieved (Anon, 2008).
Ney and Partners consulted Chris Williams on the design of a glass roof over courtyard of the Maritime Museum in Amsterdam. Its grid layout follows the axes of an old maritime map in plan view, which resulted in a mixture of triangular, quadrangular and polygonal glass panels. Here, the distinct grid pattern was prioritised over the surface geometry, and as such it is the first example of a real building where the three-dimensional shell form was found whilst respecting a requirement of the irregular glass panels to remain planar (Adriaenssens et al., 2009).

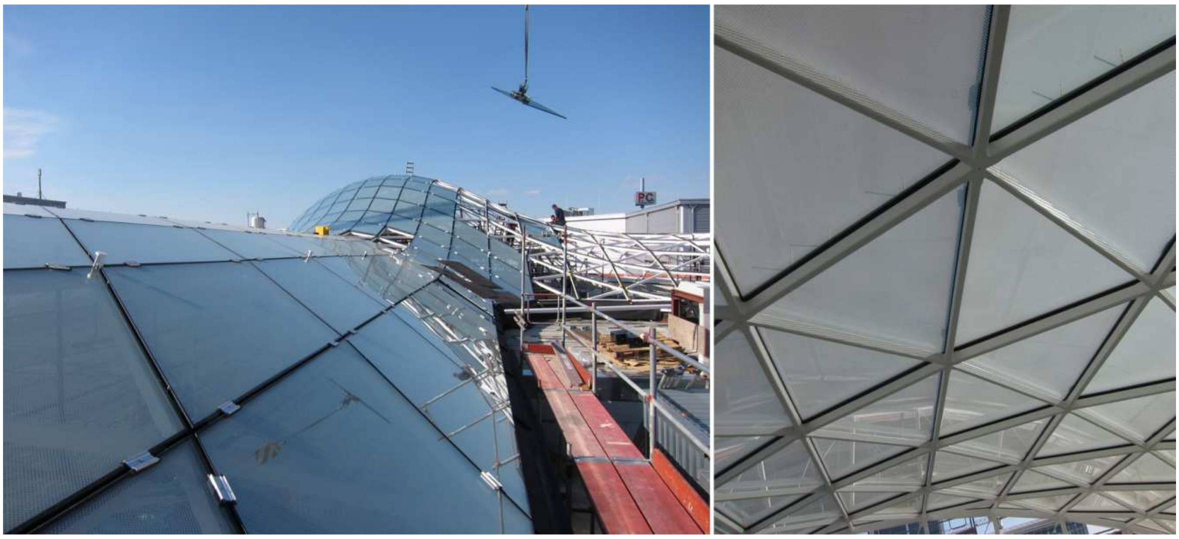

Figure 3 Building envelopes. Left: Translational surface from quadrangular glass panels. Right: Free-form surface built from triangular glass panes. Images courtesy of Schlaich Bergermann and Partners, co-designed by the author.

1.2.5 Existing software tools

The increased interest in the top-down approach to free-form design has only been met hesitantly with the development of software tools for surface restructuring. Evolute GmbH have released tools for quadrangular remeshing as a plug-in to Rhinoceros 3d (McNeel, 2011). Their tool is tailored to the architectural sector by permitting field-specific boundary conditions and the planarisation of panels.

Research cluster INRIA is developing the comprehensive open source mesh processing suite Graphite (INRIA, 2010) which is currently at an early alpha stage but offers an array of triangular and quadrangular remeshing tools, including periodic global parameterisation (see chapter 4). The software is designed to be used in computer graphics, which makes it difficult to impose appropriate architectural constraints.

Later in this thesis, surface subdivision is suggested for the creation of base complexes and their refinement. Subdivision modellers are popular in animation and computer graphics. Many polygonal modelling packages such as 3dsMax (Autodesk, 2012a), Maya (Autodesk, 2012b) or the open source tool Blender (2012) offer subdivision modelling. These are seldom designed for accuracy and necessary tools such as an intuitive snap functions are often not available.

2 Software development

To address this patchy availability of surface restructuring software several digital design tools were implemented by the author and combined into a unified design workflow. With specific architectural considerations in mind, tools for triangular and quadrangular remeshing have been developed. These have been complemented by optimisation routines using dynamic relaxation, and surface subdivision tools for the intuitive generation of geometries in the first place.

2.1 Software structure

For effective use in practice, the software tools described in subsequent chapters were developed to interface seamlessly with an existing modelling framework to ensure an intuitive, one-stop workflow. McNeel's Rhinoceros 3d is a 3d-modelling package favoured by many architects for its accurate handling of NURBS-surfaces and simplicity of use. It also offers native triangle and quad-meshes, though mesh tools are limited in comparison with other software products. Accessible via a partially open-source software development kit called Rhinocommon, it has attracted a large number of third-party plug-ins.

Grasshopper (McNeel, 2012) is one such plug-in to Rhinoceros 3d and allows users to intuitively define parametric design problems using a graphical programming interface. The user specifies dependencies between geometric elements rather than generating a one-off model. Parameters affecting the indicated relationships can be adjusted at a later stage of the design. Grasshopper interfaces with and enhances the features present in Rhinocommon.

The software is programmed in C# and interfaces with Grasshopper, where individual tools appear as so called "components" in Grasshopper's graphic user interface. At the same time, reliance on the existing programming framework is limited. In essence, the program only makes use of Grasshopper's interface and Rhinocommon's display pipelines and basic vector classes. It could therefore be ported easily to other software, or function as a stand-alone package should it be undesirable to use Rhinocommon on a given project.

Many remeshing problems can be restated as mathematical optimisation problems and must be solved numerically. To achieve this, custom Matlab (MathWorks, 2012) functions are called using a .NET-COM interface. The problems are generally sparse and large-scale matrix operations and must therefore be formulated concisely and data stored efficiently so as not to exceed limitations on memory. All matrices are represented in a sparse format, where, for each nonzero entry of a matrix, the value, row and column indices are stored in three respective vectors.

The modular, object oriented nature of the software permits its expansion when required. In fact, specific subclasses were added in the course of almost every project on which the software has so far been employed, indicating that the additional flexibility is of more than speculative use. Figure 4 shows the elements contained in the software.

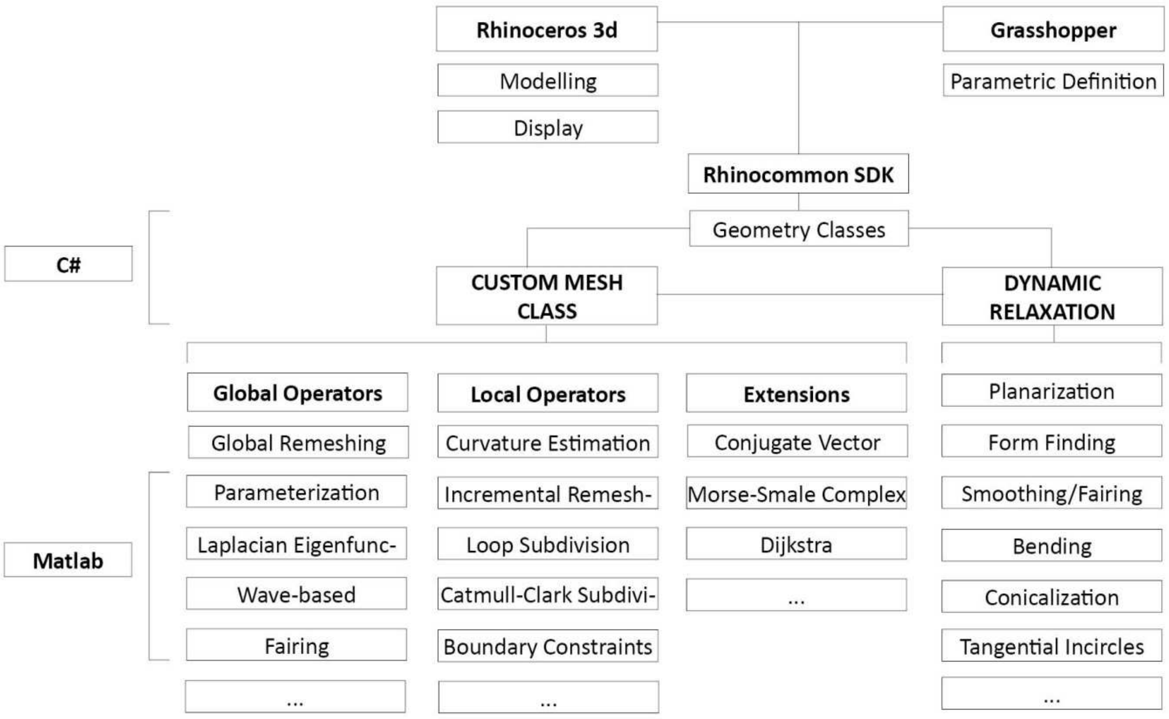

Figure 4 Structure of software.

2.2 Meshes

The software relies on triangle and polygonal meshes to represent geometries. In architecture, freeform modelling is most commonly carried out using the frameworks of solid- or mesh- modelling, NURBS or T-Spline technologies. These techniques differ fundamentally in mathematical terms, but basic modelling techniques such as extrusions or Boolean operations exist in most frameworks.

NURBS provide a powerful tool for the creation of abstract free-form surfaces as their differentiability makes the extraction of geometric information easy. Their uv-parameterisation however does not generally correspond to the topological structure of the surfaces created, and subsequent discretisation based on the extrinsic parameterisation can be counter-intuitive. Intrinsic surface properties are better suited for such tasks. Even in NURBS or T-Spline applications, meshes are used for display to better suit contemporary graphic card architectures. Meshes also form the basis of numerous finite element and optimisation methods. In particular, most structural or form finding simulations are performed using meshes. NURBS must be meshed prior to simulation, which is often an irreversible process. Designing with meshes therefore provides a workflow free from tedious conversions and very lightweight storage of data.

One drawback of mesh modelling is the loss of information when storing only discrete samples of data. However, methods exist to construct faithful estimates of curvatures, derivatives or other lost information from the sampled data (Meyer et al., 2002).

Mesh construction and terminology

A mesh M comprises geometric and topological information. In this thesis, the topological component is represented by a graph of its vertices $V = [v0, \dots , vn]$, connected by edges $E = [e0, \dots , en]$ and faces $F = [f0, \dots fn]$.

To embed the mesh in space, every vertex v_i is associated with a coordinate p_i in$\mathbb{R}^3$.

The number of vertices, faces and edges are always related by the Poincaré-Hopf formula

$$V - E + F \ = \ 2(1-g),$$

where g is the genus of the surface (Botsch et al., 2010). From this relationship a regular valence of 6 for vertices in a triangle mesh, or 4 for vertices in a quad mesh can be deduced. Vertices with valence differing from said values are called irregular.

At the core of the software is a custom winged-edge (Botsch et al., 2010) mesh class which efficiently stores topological information and has been tailored to be able to perform elementary local operations such as face addition, removal or mesh joins. Unlike the mesh class already present in Rhinocommon, it allows for polygonal faces with more than four edges. Meshes can be embedded with data globally, or locally in each vertex, edge and face. Using object-oriented programming, each edge locally stores its start and end vertex, whilst faces store three or more vertices. Every vertex v_i can return touching edges $E_i$and faces $F_i$in anti-clockwise order around its normal vector$\vec{N}_i$. Its neighbouring vertices V_i can be deduced accordingly.
The basic topological information is augmented with discrete differential operators, enabling the approximation of intrinsic geometric properties, such as derivatives, mean- and Gaussian- curvatures, or principal curvature directions. These properties are estimated at vertices and interpolated linearly across mesh faces, in effect reconstructing the properties of a smooth surface from a finite set of samples, as described in Appendix A.

3 Remeshing

The term remeshing denotes the act of approximating a given mesh with a new mesh of superior application-specific quality criteria. Such criteria include but are not limited to the number and even distribution of irregular vertices, equalising edge lengths or reducing anisotropy of mesh elements, all of which have their architectural benefits.
Regularity of elements is considered advantageous, both aesthetically and often structurally. It is also advantageous in terms of manufacture, as waste (e. g. off-cut) when fabricating panels can be reduced. Irregular nodes can be visually exciting when correctly and moderately placed, but intrusive otherwise. An automated remeshing procedure must provide the designer with control over all such aspects of the design.

Remeshing has also been widely used for surface reconstruction, mesh generation from 3d-scans, improvement of meshes for finite element calculations, mesh simplification and parameterisation for texture mapping. Often however, boundary constraints differ from the architectural setting. Recent overviews of remeshing and its applications can be found in Botsch et al. (2010) and Hormann et al. (2008).

3.1 Remeshing from a base complex

To restructure an input mesh, a global parameterisation is sought. Since not every surface can be parameterised globally, the mesh is first segmented into sub-domains, called charts. Before this parameterisation process which is outlined in chapter 4, vertices that are at once part of the input mesh and the base complex are identified.
The use of a simplified base complex for remeshing ensures that irregular vertices can only occur at nodes which are equally part of base complex and input mesh. Isotropic elements in the base complex generally lead to an isotropic segmentation of the input geometry. As such, base complexes constitute a useful tool to attain control over irregular vertex placement and to reduce the number of irregular vertices in the output mesh. A base complex can be represented by a much coarser version of the input mesh and can reveal very distinctly core topological information. Consequently, an approximation of the input mesh with reduced vertex count is sought. Once a base complex is found, the mesh can be decomposed into subdomains and parameterised globally to yield a mesh of higher quality, as shown in chapter 4. Figure 5 illustrates the process of remeshing from a base complex. The vertex count of the input mesh (left) is reduced while retaining geometric featured. The areas of the input mesh assigned to each triangle of the base complex (middle) are then parameterised and subdivided to give the final, isotropic, mesh (right.

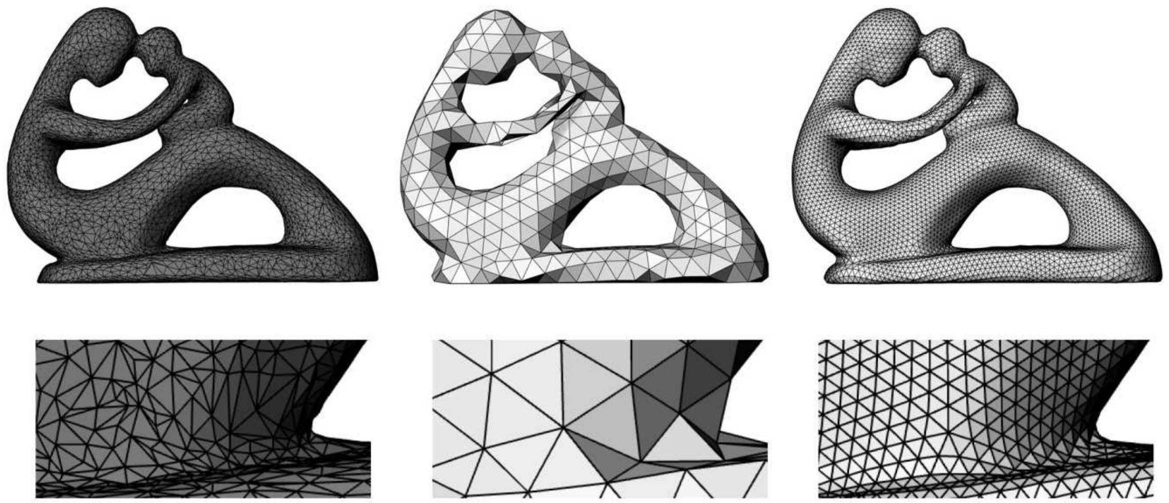

Figure 5 Left: "Badly" structured input mesh at high resolution. Middle: Decimated model to be used as base complex for remeshing. Right: Remeshed model, exhibiting only the irregular vertices of the base complex. The curvature dependent decimation ensures irregular vertex alignment with geometric features. Input model courtesy of the AIMShape repository.

3.1.1 Decimation

For triangle meshes, the base complex can be user-generated or attained by the reducing the vertex count of the input mesh to a low number, a process referred to in this context as decimation. The automated decimation can be achieved using incremental remeshing, where a number of local operations can serve to reduce vertex count while retaining key geometric features.

As this decimation process discourages anisotropy of base complex elements, the elements of the final mesh are usually of low distortion also.

Incremental remeshing can be used to generate topological base complexes, i.e. coarse versions of input meshes. The remeshing steps outlined above are performed until a desired target count of vertices is achieved. Note that because vertices are eliminated in reverse order of geometric importance, vertices of the coarse mesh will be aligned with features of the input mesh. **Fehler! Verweisquelle konnte nicht gefunden werden.**Figure 5 shows the process of base complex creation and subsequent parameterisation to create an isotropic input mesh.

3.1.2 Incremental Remeshing

Incremental remeshing performs a number of local operations to gradually improve mesh quality. The software implemented by the author follows Botsch et al. (2010) to include local operations such as edge collapses and splits, edge flips to equalise vertex valence and vertex relocation using constrained Laplacian smoothing, whereby a vertex v_i is places at the barycentre of its neighbours N_i but projected back onto its original tangent plane. The individual operations are shown diagrammatically in Figure 6.

Though remeshing generally the results in low-distortion faces, incremental remeshing alone has the disadvantage of giving little control over the number and location of irregular vertices. As such, the attained mesh quality may not be adequate for the architectural task. Incremental remeshing is however useful to generate highly isotropic meshes, which increases reliance and performance of other processing steps, in particular curvature estimates. When permitted or required by the task, the input mesh is improved by incremental remeshing before other operations are carried out.

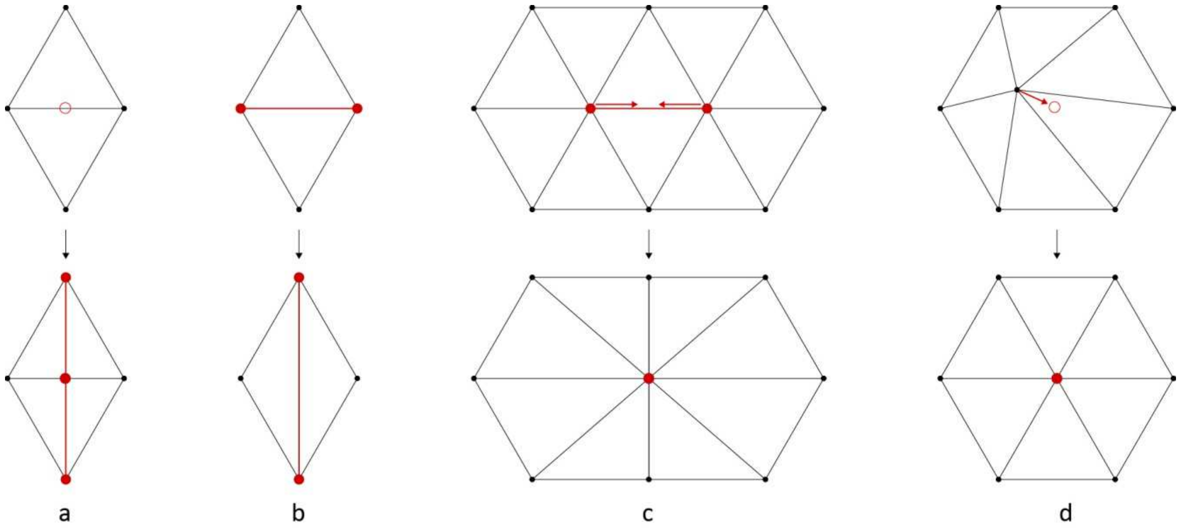

Figure 6 Incremental remeshing operations. a) Edge split. b) Edge flip. c) Edge collapse. d) Vertex relocation.

To achieve this, mesh vertices are ranked in order of the curvature estimated at their location, and vertices in areas of lower curvature are collapsed first. Similarly, edges are ranked according to their nonconformity with a desired length, and split or collapsed accordingly. For triangular meshes, the desirable valence is generally six, for quadrangular meshes four. Edges are flipped if the results leads to an overall increase in regularity of the vertices affected. Optionally, local constrained Laplacian smoothing is performed following each collapse operation. As such, a mesh vertex v_i is relocated to the barycentre of its neighbours V_i and projected back onto their initial tangent plane to avoid large deviation from the input mesh.

As suggested by the name, all operations are performed incrementally until they no longer improve on meeting specific target criteria, such as desired edge lengths.

Specific edges of the input mesh can be tagged as feature edges either by user input or by the angle between their adjacent face normals. Edges running into such feature edges are generally not collapsed to retain a good approximation of the initial geometry. This is also the behaviour sought for boundaries and creases.

4 Global parameterisation and mesh extraction

4.1 Parameterisation

The vertices of the base complex could now simply be connected to generate a very coarse base mesh, as has been done in Figure 2 (centre) and Figure 5 (centre). Since this representation is generally too coarse, these meshes must be subdivided until a desired resolution is reached, all whilst approximating the original geometry. Subdividing and projecting new vertices onto the old mesh along the surface normal may lead to inadequate results in areas of high curvature or overlapping mesh geometry, as the correct target location for the projection maybe ambiguous.

Therefore, instead of deriving the final mesh from the coarse mesh, the topological information of the base complex is used to drive a global parameterisation. Since only exceptional geometries which are not topological disks (i.e. with no or more than one continuous boundary) can be parameterised without overlaps, the input mesh is segmented into sub-domains. Every face of the input mesh is associated with a unique sub-domain stipulated by the base complex. These sub-domains, called charts, are topological disks and can therefore be individually embedded into a 2-dimensional parameter domain. To achieve a low distortion parameterisation, faces are assigned charts such that the parameter domain is already closely approximated.

When mapped back onto input mesh, two neighbouring charts always share a common edge. Iso-parameter curves translated from the parameter space therefore line up at their boundaries, and a continuous, global parameterisation is achieved. The final mesh edges are aligned with the iso-parameter curves and are therefore continuous.

4.1.1 Chart generation

Faces of the input mesh are assigned respective charts. In order to do so, the shortest path (see 4.1.2) is computed between every vertex pair associated by the base complex. Mesh edges belonging to these paths are tagged. To extract the mesh faces belonging to each chart, all mesh faces are placed into a queue. Starting at a random face, all faces which do not share a tagged edge are assigned to the current chart and removed from the queue. The assignment procedure is repeated iteratively for faces newly added to a chart until it is entirely bounded by tagged edges. Another random start face is then selected from the queue and the procedure repeated until all mesh faces have been assigned a chart.

4.1.2 Shortest path

To efficiently trace region boundaries, a propagating front algorithm has been implemented to compute shortest paths between nodes which are adjacent in the base complex but separated on the input mesh. Starting at a given vertex v_i a modified Dijkstra algorithm (Dijkstra, 1959) is used to find the shortest path to the target node.

Initially, all vertices are assigned with a geodesic distance value d_i of infinity and added to a queue. Only the start node is assigned zero distance and added to a set of vertices Fr, called front . For each iteration, vertices $v_i \epsilon Fr$ contained in the front update the distance value of their neighbours V_i in case it is lower than the one currently stored. The value d_j at v_j is found by adding the value at v_i and the length of edge e_{ij}

$d_{ij} = v_i + \|\vec{e}_{ij}\|$.

The vertices contained in the front are tagged as visited and removed from Fr. Instead, neighbouring vertices $v_j \in V_i$ are added toFr, and as such the front is propagated until the target node is added to Fr. Starting with the last node, the shortest path can then be traced back by always adding the vertex $v_j \in V_i$ with the lowest geodesic distance $d_j$to the path. Finally, the list of vertices in the path is reversed (Dijkstra, 1959).

Topological information available in the mesh is taken into account and only neighbouring vertices of the vertices of the current front are placed into the queue for the next iteration. This dramatically reduces the size of the queue that must be traversed at every iteration, and as such exponential improvement in computation time and memory cost over the original Dijkstra algorithm is achieved.

4.1.3 Planar embedding

In the following section, the procedure to extract a new mesh from the individual charts inherited from the base complex is outlined. The charts generated from the base complex are topological disks, i.e. they have only one continuous boundary. This permits the construction of a planar embedding.

To regularly subdivide a triangular or quadrangular chart C with low distortion, its image $C'$is created by its mapping onto a triangle or square in $\mathbb{R}^2$. The primitive shape is subdivided along the iso-parameter lines and the resulting vertices $v' \epsilon C'$ are mapped back to the appropriate coordinates on the input mesh. These new vertices are joined up to complete the remeshing process.

There is a great amount of literature on such planar embedding and low-distortion mappings of triangle meshes into planar domains as this is problem frequently encountered in texture mapping in computer graphics. Good overviews can be found in (Botsch et al., 2010) and (Hormann et al., 2008).

4.1.4 Parameterisation to primitive domains

In the case of a triangular base complex, each chart is mapped onto an equilateral triangle in $\mathbb{R}^2$. Similarly, each four-sided chart is mapped onto a unit square. The nodes of the base complex are placed at the corners of the primitive shapes. Along the boundary of quadrilateral meshes, triangular patches are sometimes encountered. Unlike the triangular domains of pure triangle complexes, these are divided into a combination of quads lined up with the neighbouring charts with triangles along the open boundary.

UV-coordinates are assigned to each boundary vertex of the chart depending on their position between nodes along the chart boundary. To distribute vertices inside the parameter domain such that low distortion is achieved, "spring" constants are assigned to each edge. By fixing boundary vertices, the following overall residual energy term can be minimised

$$E = \frac{1}{2}\sum_{i=0}^{n} \sum_{j \in V_i} \left(\frac{1}{2} D_{ij} \parallel u_i - u_j \parallel\right)^2,$$

where D_{ij} is the spring constant assigned to edge e_{ij}, and u_i are the unknown parameter coordinates of interior vertices (Hormann et al., 2008).

Each interior point is the affine combination of its neighbours,

$$u_i = \sum_{j \in V_i} \lambda_{ij} u_j ,$$

where $\lambda_{ij} = \frac{D_{ij}}{\sum_{k\in Vi} D_{ik}}$ corresponds to the unitised per-vertex cotangent weights derived in chapter 4.1.5.

Two linear systems can then be solved
$AU = U' \; and \; AV = V'$
where U and V are the unknown u and v coordinates in parameter space respectively. U' and V' represent known u and v coordinates of boundary vertices and A is the cotangent form of a geometry aware Laplacian matrix.

4.1.5 Geometry-aware discrete Laplacian

The graph Laplacian, is an $n \times n$ matrix describing topological adjacencies in a graph, where n is the number of vertices in V. Each connection (or, in this case mesh edge) of the graph is assigned a weight. Diagonal entries are assigned the sum of the weights of incident connections.
Often uniform weights are used (Taubin, 1995), but in geometry it is desirable to embed geometric information into the weights of the connections of the graph.

Therefore "cotangent weights" defined by
$D_{ij} = (cot\alpha_{ij} + cot\beta_{ij})$
where the angles α and β are the angles opposing edge e_{ij}. These are used to find function values for each vertex v_i
$\Delta f(v_i) = \frac{1}{2A_i} \sum_{v_j\epsilon V_i} D_{ij}(f_j - f_i).$
A_i is the Voronoi area assigned to v_i. This area describes a domain connecting the circumcentres of the triangles F_i adjacent to v_i, which results in a perfect tiling of the mesh surface (Meyer et al., 2002), as can be seen in Appendix A. Area A_i is adapted in special degenerate cases such as obtuse angles. Note that the graph loses its initial symmetry when being area weighted. Also, cotangent weights can be negative which can lead to degeneracies in the results. Depending on the application, it is therefore advantageous to use different weighing schemes, also based on curvature estimates. The software implemented offers mean-value weights as an alternative choice.
In the absence of a proper definition of the discrete Laplacian on mesh boundaries, the cotangent term of the "missing" triangle is omitted, corresponding to a Neumann boundary condition (Vallet and Lévy, 2008) .

4.1.6 Mesh extraction

In the subsequent reversal of the process, vertices of the desired mesh are mapped back onto the original triangle mesh. For each vertex v_i' in $\mathbb{R}^2$, the surrounding mesh face F' of the embedded mesh C' is found, as are the barycentric coordinates of the vertex with respect to F'. We then create a vertex v_i of our final mesh in the corresponding coordinates of the corresponding face F of the input chart C.

Results of the remeshing procedure from a base complex can be seen in Figure 2 (right) and Figure 5 (right) as well as in the case study in chapter 0.

4.1.7 Globally sooth parameterisation

An elegant method for simultaneous parameterisation of multiple triangular charts has been proposed by Khodakovsky et al. (2003). The method maps each chart to an isosceles triangle but re-assigns vertices to adjacent charts if this reduces distortion. This is particularly useful in case the base complex returns non-bijective charts.

The globally smooth parameterisation method has subsequently been adapted to quadrilateral charts by Dong et al. (2006). Similar the above method, vertices are assigned to charts if required, but the fact that four-sided quads can be rotated with respect to another adds complexity to the problem.

Globally parameterizing quadrangular charts therefore requires the introduction of rotational invariance when re-assigning vertices. Zhang et al. (2010) further reduce distortion by mapping the charts onto rectangles rather than squares to account for anisotropy.

Test cases showed that in the architectural setting, such globally smooth parameterisation was not necessary, as many of the degeneracies in the base complex came from intricate features in the geometry such as rings, which are less likely to occur in the comparatively simple geometries occurring in architecture.

5 Quadrilateral remeshing

Translated into architecture, quadrangular meshes exhibit numerous advantages over their triangulated counterparts. Nodes combine four rather than six edges, making their detailing simpler, both visually and in terms of manufacture. A greater surface-to-structure ratio gives them a lightweight feel. When used for shell structures, they must however be braced diagonally.

Generating a quadrilateral base-complex for remeshing can be more challenging but the principles of remeshing from a base complex can be carried over from triangle meshes. As before, a basic representation of the input surface is sought to guide a global parameterisation of the input mesh.

Before processing to the creation of quadrangular meshes, several criteria for the pursued mesh quality and layout are defined. Following an appraisal of literature, spectral graph theory is used to place a periodic function on a mesh. The construction of a base complex from such a function is outlined thereafter. Following on, it will be shown how the orientation of the mesh elements can be controlled by a guidance vector field. This requires tools for their definition and visualisation of triangular meshes. These are visualised and used for either manual or automatic parameterisation.

5.1.1 Quad quality

While it is of course trivial to turn a triangulated mesh into a quad mesh by connecting any point inside each triangle to the edge midpoints of the face, quad remeshing addresses at once requirements to size and quality of the individual quads and of the overall topological structure of the resulting mesh. Specific, mathematically quantifiable, goals as to the mesh quality are therefore defined.

Alignment with architectural design constraints

Constraints imposed on architectural remeshing are often more stringent than those encountered in computer graphics. Exact node placement may take precedence over quad quality, as irregular vertices or other nodes can be visually intrusive and may need constraining to specified locations. Structure and boundaries must be resolved in ways which are aesthetic and aligned to structural supports alike.

Regularity and features

A successful quad mesh needs to retain features of the input mesh at the same time as attempting to reduce the number of irregular nodes. Ideally, these are spread out regularly over the mesh and occur in areas of geometric significance, or where they are unobtrusive to the design.

Anisotropy

Most applications seek quad meshes where the orthogonality of angles is maximised. In computer graphics this reduces distortion of texture maps (Levy et al., 2002). The accuracy of finite element simulations is generally improved when meshes are regular and isotropic (Bommes et al., 2012).

In architecture, mostly square or rectangular quads are generally considered aesthetically pleasing. In façade manufacture they further minimise off-cut when producing cladding panels. Near-orthogonal angles also facilitate the manufacture of steel nodes as acute, difficult to weld, angles are avoided.

Accurate surface sampling is perhaps the exception to these considerations, because anisotropic meshes can lead to tighter error bounds with respect to deviation from the sampled surface (Kovacs et al., 2010).

Planarity or single curvature

The built environment is intrinsically tied to the material world which is why manufacturing considerations are ideally embedded into the process of mesh creation. Planarity or near-planarity is one such criterion, in particular when working with sealed, multi-layered glazing or other composite panels. Others, such as the restriction to singly curved panels or circular arc structures have also been addressed (Bo et al., 2011).

Alignment with curvature or other conjugate vector fields

For every point on a surface, there is a set of two orthogonal principal curvature directions which generate a conjugate vector field, with the additional property of being orthogonal everywhere (Pottmann, 2008). Interestingly, the streamlines of the principal curvature direction field are the ones an artist would intuitively draw to represent depth on a curved object. This “natural” correlation makes it very useful for architectural remeshing as it permits an “honest” reading of the intrinsic geometry properties. The singularities of the curvature field occur at the umbilical points of the surface. Quadrangular panels whose edges follow principal curvature directions are often near-to planar (a small deviation from planarity can occur due to the discrete sampling), making them suitable for subsequent optimisation.

Recent methods also permit the definition of a user-specified vector fields, drawn onto the surface either manually (Liu et al., 2011, Ray et al., 2008) or aligned with other fields such as principal stress (Schiftner and Balzer, 2010).

5.1.2 Literature

Quad remeshing has been endowed with great amounts of research over the last decade. An elaborate review of state-of-the-art methods can be found in (Bommes et al. 2012).

Spectral methods

Spectral mesh processing methods were first used by Dong et al. (2006) to derive quadrangular base complexes for quadrangular remeshing. Huang et al. (2008) improved the method by allowing for alignment control. A spectral remesher was implemented and a more elaborate overview is given in chapter 0.

The alignment control offered by spectral methods is very limited. To address this problem, methods have been put forward which use conjugate vector fields defined on the input mesh to guide remeshing.

Streamline tracing

Alliez et al. (2003) trace the streamlines of a smoothed principal curvature vector field in a planar embedding of the mesh and use the intersections between the respective lines following the maximum and minimum directions to generate a quad mesh. A roughly uniform distribution of streamlines is achieved by placing potential seed points for new lines at a radius determined by the local curvature either side of the currently traced lines. This point seeding technique is used in chapter 8.2 to visualise streamlined on a mesh. Marinov and Kobbelt (2004) extend this method by tracing curvature lines directly on the mesh surface. The method avoids the necessity for a parameterisation of the mesh.

Periodic global parameterisation using guidance vector field

Ray et al. (2006) propose a global periodic parameterisation whereby a globally periodic wave function is aligned with the principal curvature field. The iso-parameter lines of the wave function generate a smooth and continuous rectangular grid following the principal curvature lines when parameterised. Their parameterisation method automatically detects singularities and cuts the mesh into quadrilateral charts defined by the vector field separatrices. Later on, Ray et al. (2008), explore the underlying rotationally invariant vector fields more rigorously.

Zadravec and Schiftner (2010) use a similar method to extract quad meshes but provide additional flexibility by permitting additional singularity types when generating a guidance vector field. Liu et al. (2011) also use rotationally symmetric vector fields to generate near-planar quad meshes.

Wave-based quadrangulation

Zhang et al. (2010) place a periodic standing wave on the surface. Similar to the global periodic parameterisation proposed by Ray et al. (2006), vertices are assigned parameter values of a two-dimensional periodic wave function, which are subsequently optimised to align with an input guidance field. The iso-parametric curves of the resulting function divide the mesh into a set of quadrilateral domains which are extracted to form a quad mesh using a Morse-Smale Complex (see chaper 0).

Branched coverings and quad domains

Branched coverings are used by Kälberer, Nieser and Polthier (2007) to parameterise surfaces according to conjugate vector fields. The individual directions of the conjugate input field (i.e. maximum/minimum principal curvature vectors) are treated as separate layers to contrive a base complex. The concept is extended to the volumetric parameterisation of closed volumes in $\mathbb{R}^3$ by Nieser, Reitebuch and Polthier (2011). Volumetric parameterisation could in the future become a very exciting field for architectural design, as it could provide the mathematical formulation for the creation of orthotropic trusses and space frames which are “naturally” aligned with their envelope, bridging the conceptual gap between structure and surface.

6 Quadrangular base complex using spectral methods

The first quad-meshing method implemented as part of the software discussed in this thesis relies on spectral quadrangulation. Although it does not allow for alignment control, it is a mathematically elegant way of segmenting sufaces into topological disks, producing regular and isotropic quad meshes over topologically complex meshes. The method is based on notions from spectral harmonics, which must be reviewed first.

6.1.1 Spectral harmonics

Spectral processing methods allow for the definition of a natural frequency domain over triangle meshes (Levy and Zhang, 2010). Botsch et al. (2010) give an intuitive derivation of manifold harmonics as the natural extension of the Fourier series onto discrete manifolds. Analogous to the discrete cosine transforms on a 2d-plane, the mesh is reinterpreted as a summation of periodic wave-functions at increasing frequencies. These frequencies, the *spectrum* of the mesh, correspond to the eigenvalues of the Laplacian graph, which charts the mesh's connectivity. Their corresponding eigenvectors can be understood as discrete samples of the Laplacian eigenfunctions of a mesh (Taubin, 2000). As such, a triangle mesh can be described as the composition of the individual eigenfunctions of its so called *manifold harmonics basis* (Vallet and Lévy, 2008).

One of the properties of such functions is that their extremes are periodically distributed. Augmenting the graph Laplacian with curvature information can result in functions whose singularities tend to align with geometric features.

By representing the geometry using only a selected band of the mesh spectrum, spectral harmonics have previously been used for geometry fairing (Taubin, 1995). Similarly, spectral methods from image compression have been adopted to efficiently compress triangle meshes (Karni and Gotsman, 2000).

6.1.2 Spectral quadrangulation

Laplacian eigenfunctions generate periodic scalar fields over the mesh surface. Such functions are periodic over the mesh surface with wavelengths corresponding to the chosen eigenvalue. This periodicity in orthogonal directions has been taken up to generate quadrilateral base complexes on triangle meshes.

A Morse-Smale complex (see chapter 0) divides any scalar function defined on a surface into quadrangular patches (Edelsbrunner et al., 2001). By applying such a base complex to Laplacian eigenfunctions, Dong et al. (2006) propose a very elegant method for pure, isotropic quad meshes with only few irregular vertices. Their method does however not permit the alignment with an input vector field. Huang et al. (2008) improve on this method by permitting weighted orientation and alignment control to a degree.

6.1.3 Spectral quad mesher

The Laplacian graph (see chapter 4.1.5) constitutes the base of spectral mesh processing. Cotangent weights D_{ij} are used to generate the normalised Laplacian matrix

$$L = \begin{cases} 0 & if\ j\ \notin V_i \\ 1 & if\ j = i \\ \dfrac{-D_{ij}}{\sum_{k\ \epsilon V_i} D_{ik}} & if\ j\ \in V_i \end{cases}$$

Following the steps outlined in the same chapter, eigenvalues and corresponding eigenvectors are extracted from this matrix to derive discrete samples of the Laplacian eigenfunctions. As shown, these functions describe two orthogonal wave functions on the surface.

Using Matlab, the computation of values is restricted to a desired sub-set of the mesh spectrum. Eigenvalues are real and positive and range from 0 to 1 (Dong et al., 2006). Higher eigenvalues exhibit larger amounts of singularities and shorter wavelengths. By interactively plotting the pre-computed eigenfunction values as a colour graph, the user can quickly cycle through the results to select a function of suitable wave length and singularity placement. For certain frequencies, singularities align with geometric features and are otherwise distributed evenly over the mesh (Dong et al., 2006).

In general, a fairly low eigenvalue is sought to reduce the number of singularities and corresponding irregular vertices in base complex and final mesh.

6.1.4 Results

Figure 7 shows Laplacian eigenfunctions displayed in false colours on the “rocker” model. The frequencies get progressively larger from left to right, reducing the wavelength of the functions. Note that at this point a periodic wave function has been defined a on the input mesh, but a quad mesh has yet to be extracted. To extract an all-quad mesh, a Morse-Smale complex is computed.

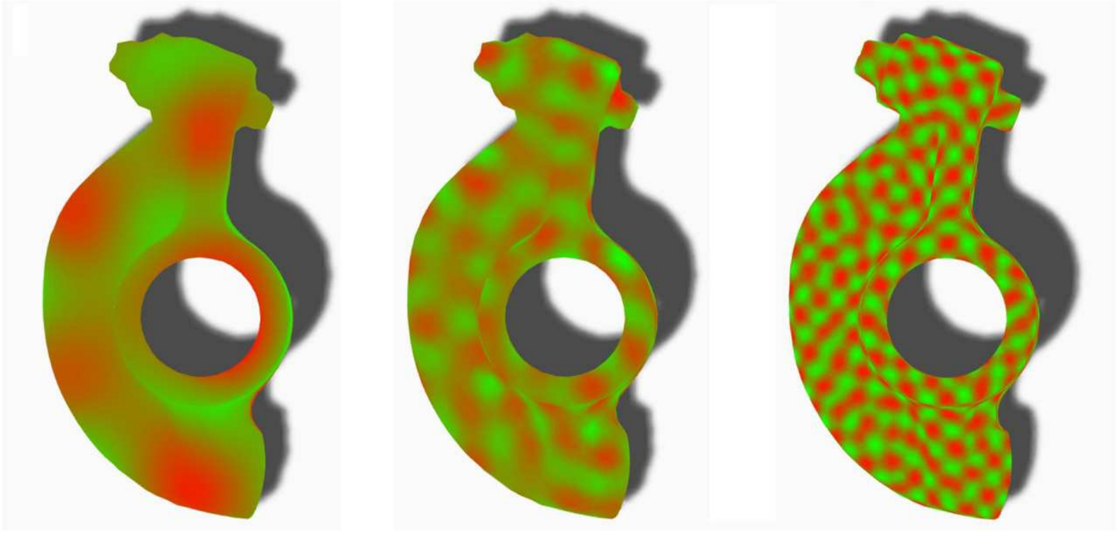

Figure 7 Eigenfunctions of varying frequencies on an input mesh. Increasing eigenvalues (from left to right) reduce the wavelength.

7 Base complex from scalar periodic function

In the previous chapter, a discrete periodic function has been defined over the mesh. Once such a function has been applied to the mesh, a base complex still needs to be constructed to guide desired global parameterisation and remeshing.

To construct a quadrangular base complex, a tool from homology theory called a Morse-Smale complex is used. This complex connects saddles with the extreme points of scalar function values defined on the surface. In the previous chapter, one such function has already been computed and applied on the mesh. For periodic functions, the Morse-Smale Complex leads to a decomposition of the input mesh into regular quadrangular charts. The individual charts are topological discs and can be parameterised and subdivided to yield the final mesh.

7.1 Morse-Smale Base Complex

7.1.1 Intuitive definition

Imagine an undulating or roof surface, rain falling onto the roof will flow downward until it hits a collection point located below. Water falling onto the roof will always follow the same locally steepest downward path and split at the same ridges. Consequently, every sink will collect water from a distinct region. Equally, if gravity were reversed with rain coming from below, water from a certain region would collect at always the same apex of the roof. Similar to a watershed, a Morse-Smale complex can be used to decompose the roof into individual regions associated with a sink or apex each.

7.1.2 Formal definition

Expressed in a more formal way, a (differentiable) scalar function f defined over a 2-dimensional domain D will exhibit a gradient field of vectors pointing into the local direction of steepest ascent.

A Morse-Smale complex decomposes this vector field into regions of uniform flow. Each of these regions, called *charts*, will be bound by a minimum, a saddle, a maximum and yet another saddle, and will therefore be a topological quadrilateral (Edelsbrunner et al., 2001).

To compute the Morse-Smale complex in practice means determining saddle points of the height function and associating them with maxima and minima in their vicinity by tracing the lines of steepest ascent and decent until an extremum is reached (Edelsbrunner et al., 2001). Vertices associated with another in this way are called Morse-neighbours.

In the discrete setting, locations of function maxima and minima are determined by testing whether the scalar function value at vertex v_i is higher than those of all neighbouring vertices V_i.

To identify saddles the change in gradient of the function values when going around the one-ring V_i of v_i is examined. The emanating edges pointing in directions of locally steepest ascent and descent are registered. A regular saddle exhibits four interwoven maximum and minimum directions. However, a saddle can be of greater valence.

Once found, directions of steepest descent and ascent are traced until a vertex with maximum or minimum function value is hit, or until the path meets another saddle.

7.1.3 Degeneracies

Various degeneracies can occur when building the Morse complex of an arbitrary height function on a mesh (Edelsbrunner et al., 2001), in particular if the function behaviour is irregular or of short periodicity.

Tracing the directions of steepest ascent commonly leads to paths merging before arriving at local extremes. The shared boundary leads to non-bijective charts where one of the patch extremes is connected to the patch only by edges. In this case the shortest path is computed in place of the path of steepest ascent. In case these still merge, the local vertex valence is increased by local subdivision and vertices visited by the first path are tagged to be ignored when computing the second.

Noise and numerical instabilities in function f can lead to the identification of superfluous extremities, often occurring in clusters (Edelsbrunner et al., 2001). Another frequent degeneracy is an ascent line reaching a saddle before it reaches a maximum. Post-processing steps are therefore applied to the complex to ensure its validity.

Similar to the edge collapse procedure in incremental remeshing, degenerate saddle-saddle pairs are removed by collapsing them into one and computing new patches to each of the previous saddles Morse neighbours.

To remove extraneous critical points, cancellations are performed (Edelsbrunner, Harer and Zomorodian 2001). Saddle-extremum pairs are ranked by the difference in their function values. In case they fall under a user-specified threshold, the pair is collapsed, and new connections are created between their previous Morse-neighbours. Since steepest ascent lines are no longer valid, the shortest path is computed. Vertex pairs are cancelled until a desired number of extremes is reached.

7.1.4 Quasi-Dual

The "quasi-dual" Morse complex can be derived from the initial Morse-Complex. By omitting all Morse saddles but instead connecting their Morse neighbours, another quadrangular complex is obtained. Care needs to be taken for multi-valence saddles as described in the previous section as they generate polygonal patches with more than four vertices. Since their valences are also even, however, trivial diagonals through the polygonal patch can turn them into quadrilateral domains.

The quasi-dual contains only about half the points of the primal complex and can therefore lead to a reduced amount of irregular vertices in the final quad mesh (Dong et al., 2006).

7.1.5 Mesh extraction

Once the Morse-Complex exhibits no further degeneracies, chart boundaries can be traced and faces assigned in the same manner as for the triangular base complexes in chapter 4.1.1.

7.1.6 Results

In Figure 8, results of spectral quadrangulation on a simple spherical input mesh are illustrated. First, an appropriate eigenfrequency is chosen according to the desired number of irregular vertices. The Morse-Smale Complex then segments the mesh into sub-domains which in turn are parameterised and subdivided.

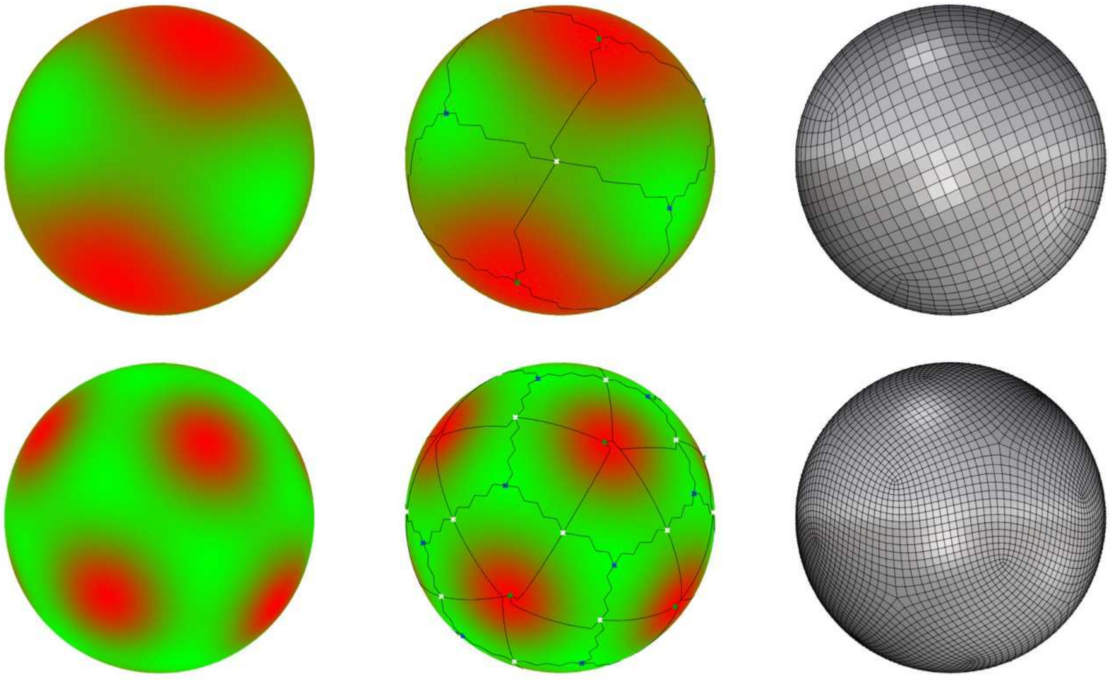

Figure 8 Spectral quadrangulation steps on a sphere model, using the 10th (top) and 26th eigenfunction (bottom). Left: the wave function at varying frequencies. Centre: A Morse Smale complex decomposes the input mesh into quadrilateral domains. Right: The subdivided output mesh. Irregular vertex locations and valences correspond to maxima and minima in quasi-dual of the Morse-Smale Complex.

8 Quadrangular base complexes from guidance vector fields

In practice the meshes generated using spectral methods are quite isotropic and contain few irregular vertices, but the lack of alignment control presents a major drawback. This deficiency is addressed by defining a conjugate vector field on the mesh and using it to steer the quadrangulation in the sense that any edge present in the resulting quad mesh should at its midpoint be parallel to either of the conjugate directions.

The aim is to construct a mesh whose edges are aligned with an input vector field. Also, singularities in the vector field should coincide with irregular vertices in base complex and final mesh.

In fact, the problem of defining a suitable base complex can be abstracted to the optimised placement of a wave function along a guidance vector field (Ray et al., 2006). Consequently, techniques of defining vector fields on triangulated meshes must be reviewed.

8.1 Conjugate vector fields on triangulated surfaces

8.1.1 Vector fields on meshes

A conjugate vector field defined everywhere on the mesh surface is sought.

In the architectural setting, a strongly simplified vector field may be desirable due to the correspondence of singularities with out-of-the-ordinary detailing and unwanted visual effect.

An even distribution of singularities is commonly desirable to avoid the concentration of visual artefacts.

Equally, it makes sense to locate singularities at points which by another nature have acquired special status, such as geometric or structural features or points meeting building services. Singularities close to the boundary are preferably pushed out of the domain in consideration.

The implemented program can handle different cases of vector field input. A potential field can be generated from points tagged as sources or sinks over the mesh, the gradient of which can be used as input vectors. Equally, the user may provide complete vector fields found by estimating principal curvature or stress vectors. Alternatively, if only singular vectors or guide curves are supplied, vectors of vertices at a chosen proximity of the curve are aligned with the tangent direction at respective closest points of the curves. A global vector field is then generated by extrapolating the field into the remaining areas of the mesh. This procedure can also be used to locally override an existing vector field, in which case the user chooses a weight to retain the values when subsequently smoothing the vector field.

A component to locally rotate the vectors of a field around the vertex normal by a user-specified angle was also added. Figure 9 shows an enhanced curvature field on a simple input mesh. It has been propagated into ill-defined areas (left), and locally overridden using an input curve (right).

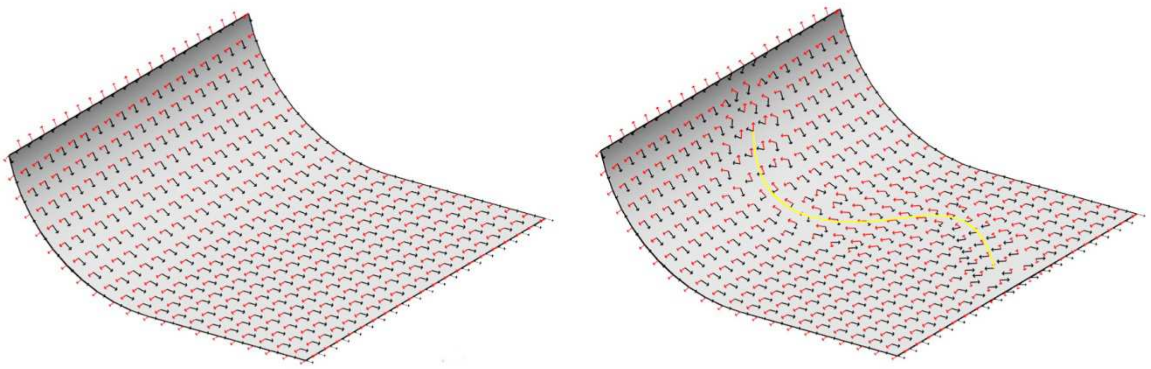

Figure 9 The vector field is carried over into usually ill-defined flat areas. Right: The vector field is locally overridden using a guide curve. Vectors in close proximity of the curve are aligned with the tangent vector at the closest point. The transition to the "normal" vector field is smooth and it's radius chosen by a user-defined parameter.

8.1.2 Literature

Hertzmann and Zorin (2000) approach the problem of creating a conjugate vector field from a computer graphic point of view when they produce cross-hatches on 3d-surfaces following smoothed principal curvature fields. They propose methods for denoising the field dependent on the reliability of the curvature measure.

In his PhD thesis, Tricoche (2002) presents techniques which extend notions from vector field interpretation and design to triangulated and quadrangulated surfaces, as well as simplically partitioned volumetric domains.

Ray et al. (2008) formulate a mathematical formulism for n-symmetry vector field design, which generalise their earlier work on rotationally invariant vector fields (Ray et al., 2006).

Bommes et al. (2009) compute a smooth symmetric cross field on a mesh and parameterise the mesh using a mixed-integer solver.

Liu et al. (2011) extend the principles of rotationally symmetric fields to conjugate direction fields which form general cross fields without the orthogonality requirement. This enhances the design freedom of placement of singularities, but it remains insufficiently intuitive for an application in architecture.

A framework for the creation of smooth vector fields in between an arbitrary, user defined number and location of singularities is proposed by Crane, Desbrun and Schröder (2010). They provide the mathematical proof which ensures no other singularities than the ones drawn by the user are present on a given surface. This framework can be constrained to smoothing cross-fields, making it suitable to quad remeshing.

8.1.3 Vector field generation

A unique pair of orthogonal unit vectors X_i and Y_i defined in the tangent plane of each mesh vertex v_i is sought. Globally, these are stored in vector fields X and Y. Each vector $X_i$is assigned a magnitude value μ_i which will be used in the anisotropic remeshing process to guide edge lengths. Similarly, each vector Y_i is assigned a length value η_i. For simplicity, these the grouped size values are referred to as the size fields μ and

η of vector field X and Y. To generate a valid field everywhere on the mesh surface, the vector field defined at the mesh vertices is interpolated linearly across mesh triangles.
Generally, unrefined vector fields exhibit degeneracies or do not comply with additional requirements such as nicely distributed singularities. The input field must therefore be refined prior to use. Also, the supplied vector field may be subject to a lot of curl which is removed by adjusting the supplied size field (see 8.1.4).
In the case there is an ambiguity in the direction of the vectors, as is the case when using principal curvature or principal stress fields, two suitable vectors of a cross field $C_i = [X_i, Y_i, X_i', Y_i']$ where $X_i' = -X_i$ and $Y_i' = -Y_i$ are selected.

When using the principal curvature field as input, isotropic regions, i.e. regions which are locally spherical, the mean curvature values cancel each other out and the directions are no longer unique. Similar to the above extrapolation, the values in such degenerate areas is overridden. Moreover, noise in the input mesh, as well as numerical instabilities when estimating principal curvatures, may lead to an undesirable number or badly placed singularities in the vector field.

Linear interpolation of field inside mesh faces

The vector fields X and Y are defined in the tangent plane of the vertices. For simplicity, the vectors defined at triangle corners are rotated into the plane of each individual triangle by
$Y_i = v_{N_i} \times X_i$ and $X_i = Y_i \times N_i$, where × is the vector cross product. This simplifies subsequent tasks to two-dimensional problems. The vector field inside the triangle is assumed to be the linear interpolation

$$X_p(x,y) = \sum_{i \in V_T}^{2} B_p(x,y) X_i$$

between the vectors defined at the corners $V_T$of the face, where $B_p(x,y)$ denotes the barycentric coordinates for each point $p(x,y)$ inside the triangle T.

Figure 10 illustrates the vectors and labels of vector fields X and Y. Note that the vectors at edge mid-points are also of interest.

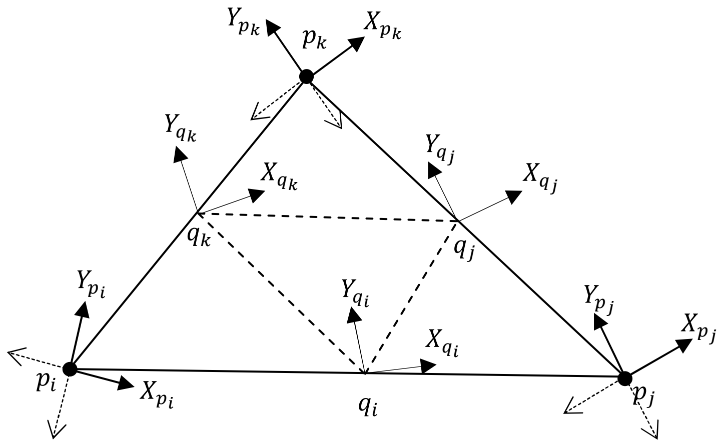

Figure 10 Diagram illustrating definition of a cross- and corresponding vector field on a single mesh face. The dashed arrows represent the discarded vectors of the cross field.

Singularities

The linear interpolation between vertices actually restricts the types of singularities that can occur inside a mesh face to a maximum valence of three. Higher valence singularities can occur at mesh vertices.
Singularities occur at locations if and only if the linearly interpolated gradients of the vector field vanish to zero at barycentric coordinates inside the mesh face.

Singularities are classified by their Poincaré index, which corresponds to the winding number for the Gauss map (Palacios and Zhang, 2007). More intuitively, this index reflects the number of rotations of the vector field in an infinitesimal counter clockwise oriented circle around the singularity. Note that for any vector field on a manifold the Poincaré-Hopf theorem $X = V - E + F$ where $X = 2 - 2g$ and g is the genus of the manifold, must hold.

The asymptotic directions of the vector field for each singularity are found using the method proposed by Tricoche (2002). In effect, the method finds the angles to those points on the triangle edges where the vector field is approaching the singularity asymptotically.
These asymptotes correspond to the sought chart boundaries and subsequent remeshing ideally places edges along the seperatrices occurring at these asymptotes (Palacios and Zhang, 2007).

8.1.4 Vector field pre-processing

Propagation and smoothing

To extend meaningful vectors into undetermined regions, a method proposed by Ray et al. (2006) is used. This method also distributes singularities evenly over the mesh.
In a two-step process, the partially supplied field is first extended into undetermined regions. In the case of principal curvatures, such unreliable areas are identified by the similarity of the two principal curvature values. Areas of near-zero curvature are also tagged unreliable. In a subsequent step, a smoothed curvature field is created using the first as starting point.

Following Ray et al. (2006), a composite error term is constructed. The first part penalises deviation from the initial vector field, whilst another measures the error of misaligned vectors. A user-selected parameter controls the weighing of either term.

The first term minimises the rotation of the smooth vector field with respect to the original angles $cosa_i, sina_i$for every vertex v_i in the plane of each triangle.

The second term seeks to construct a vector field which minimises the angle difference of the vector field in the tangent planes of adjacent vertices. To do so, the angle difference of the tangent planes must first be disregarded. A reference direction H_i is picked for every vertex by projecting the vector of an emanating edge $\vec{e}$ onto the tangent plane

$H_i = unitize(\vec{e} - (\vec{e} \cdot N_i)N_i)$

where $N_i$is the unit normal at v_i. This allows the approximation of the angle between tangent planes and subtract it from the vector field angles. The angles are then expressed as cosine-sine vectors in the triangle plane.

Adding fitting and smoothing term gives

$$R = (1-\rho)\sum_i w_i \left\|(cos\alpha_i, sin\alpha_i) - (cosa_i^0, sina_i^0)\right\|^2 + \rho \left\|\begin{pmatrix} cos\alpha_{i\oplus2} \\ sin\alpha_{i\oplus2} \end{pmatrix} - \begin{pmatrix} cos\beta_i & sin\beta_i \\ -sin\beta_i & cos\beta_i \end{pmatrix}\begin{pmatrix} cos\alpha_{i\oplus1} \\ sin\alpha_{i\oplus1} \end{pmatrix}\right\|^2$$

where $\alpha_i^0$is the original angle of the vector field at vertex v_i. $cos\beta_i = H_{i\oplus1} \cdot H_{i\oplus2}$ and $sin\beta_i = (H_{i\oplus1} \times H_{i\oplus2}) \cdot N_i$. Here, the $\oplus j$ operator denotes modulo j when moving in anti-clockwise direction around the triangle. w_i is a weighting factor used to prioritise vertex values with more robust vector field data. If the curvature field is used, $|kmax_i/kmin_i|$ can be used (Ray et al., 2006).

As vector norms are quadratic, the resulting least squares system of equations is a nonlinear minimisation problem. Matlab is used to solve.

The final vector field is found by interpolating linearly between the fitted and smoothed field. A user-supplied parameter ρ is used to determine the respective weights of the two input fields.

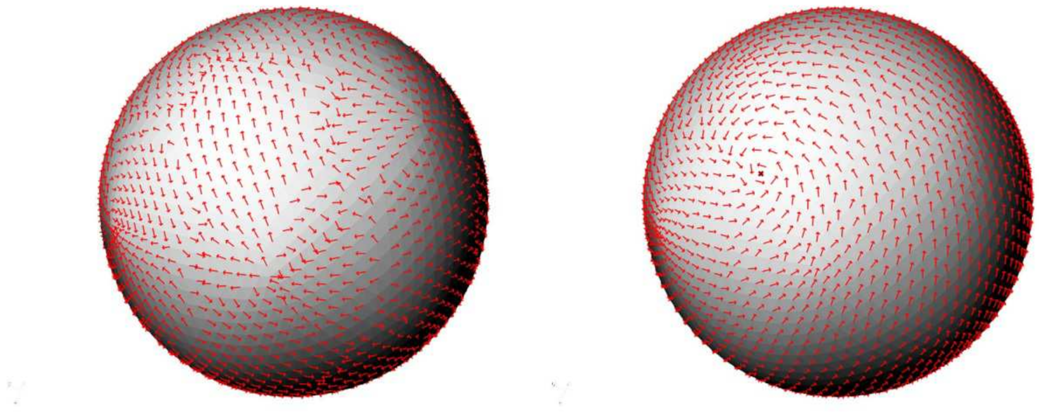

Figure 11 Vector field on sphere model. Normally, a curvature field would be ill-defined as a sphere is umbilic everywhere. Propagation and smoothing leads to a smooth vector field, as well as an even distribution of singularities.

Curl correction

Curl is a mathematical description of the infinitesimal rotation of a vector field (Ray et al., 2006). Intuitively, curl is the phenomenon by which an object will begin to rotate when exposed to fluid flow passing by, for example, a paddle wheel, free to rotate around a vertical axis, placed onto a stream of water.

Even if the conjugate size fields μ and η are kept constant at constant values, the vector field will possess considerable amounts of curl, arising from the varying orientation of vectors with respect to another. This curl leads to additional singularities in the vector field. To preserve the directions of the field, this curl is counteracted by adjusting size field values only. New magnitudes for μ and $\eta$are chosen such that curl is globally minimised over the mesh. The resultant size field is used to scale the unit vector field before applying the wave function.

As shown by Polthier and Preuß (2003) the curl in a triangle T can be approximated as

$$curl(X) \approx \frac{1}{A}\sum_{i=0}^{2} X_{q_i} \cdot (p_{i\oplus 1} - p_i),$$

where A is the area of the triangle. For this approximation to hold, the vector best aligned with X_{q_0} must be selected from the four vectors $[X_{q_j}, Y_{q_j}, X'_{q_j}, Y'_{q_j}]$ defined at each edge midpoint q_j.

In turn, the curl-free requirement can be expressed as

$curl(\mu X) = \vec{X} \cdot \nabla \tilde{\mu}$ where $\tilde{\mu} = \log(\mu)$.

From the triangle shape functions, the following relationship can be deduced

$\nabla\tilde{\mu} = \sum_{i=0}^{2} \tilde{\mu}_{q_i} \nabla B_i$,

where $\nabla B_i$are the constant vectors of the gradients of the linear barycentric coordinate functions in

$T' = \{q_i, q_j, q_k\}$, see

Figure 10. A derivation can be found in (Zhang et al., 2010).

The following energy term to minimise the curl of the vector field can then be formulated

$$E_{curl}(\tilde{\mu}, \tilde{\eta}) = \sum_{T\epsilon F}\left(\sum_{i=0}^{2} curl(\mu X) - curl(X)\right)^2 + \sum_{T\epsilon F}\left(\sum_{i=0}^{2} curl(\eta Y) - curl(Y)\right)^2$$

The solution to the above system is not unique (Zhang et al., 2010). Additional constraints to penalise deviation from designated size values at chosen edges are therefore added. Finally, to counteract a phenomenon of rapid oscillation which is frequently encountered in finite element processing, a smoothing term is added. The individual terms are weighted and added to form an $m \times n$ linear least squares minimisation which is solved using Matlab.

8.2 Streamline trace

The individual vectors of the vector field can now be visualised as little arrows using the vector preview class in Grasshopper. Displaying the streamlines of the vector field gives a more intuitive representation of its topological build-up.

A bespoke 4th-order Runge-Kutta integrator has been programmed to iteratively trace ("integrate") streamlines directly on the mesh surface. Tracing is initialised in the asymptotic directions of each singularity, and continued until either another singularity is hit, the streamline leaves the mesh or comes back on itself. The conjugate directions of the vector field are traced separately.

During tracing, potential new seed points for subsequent streamlines are placed at a user-specified distance from the current curve. Subsequent source points are automatically selected from this queue of potential seed points and deemed valid if they are not too close to an existing curve. Streamlines are traced in the four directions of the vector field. This procedure is continued until no more seed points meet the validity criterion. Streamlines can either be traced on meshes or using curvature directions on Rhino-native NURBS surfaces.

This technique of fairly regularly drawing streamlines can directly be used to generate quad meshes. By intersecting the curves of the two orthogonal principal directions, vertex locations of a quad mesh can be found. The points belonging to a quad can be joined up by simply "walking" along one of the curves from the first intersection until the next intersection is hit. Walking is then continued along the curve the intersection occurred with. Taking care to walk in the correct of two directions, the same happens at the next intersection until a full topological circle is made and the quad can be drawn.

This is a very quick method to generate anisotropic meshes, but degeneracies are frequent and must be corrected manually.

8.2.1 Results

Figure 12 shows a quad mesh generated from traced curvature lines. Valence and asymptotic directions of the singularities are clearly retained in the quad mesh. The colour coding represents estimates of the planarity of panels.

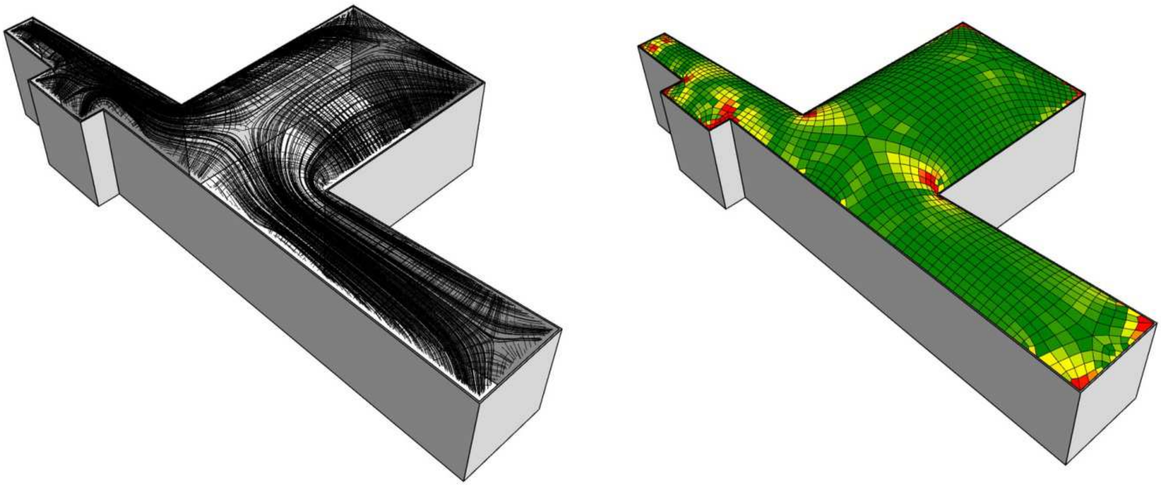

Figure 12 Left: Streamlines of the curvature field of a glass roof design. Right: By connecting intersections of maximum and minimum curvature field streamlines at regular intervals, the surface is restructured from quadrangular elements. Model courtesy of Schlaich Bergermann and Partners.

8.3 Wave-based quadrangulation

The incremental placement of streamlines can lead to degeneracies and high anisotropy in meshes. For this reason, a third quad meshing algorithm, following the method proposed by Zhang et al. (2010) has been incorporated into the software suite. In this method, the mesh layout is designed to follow the flow lines of a vector field chosen by the user whilst respecting boundary constraints and alignment to feature curves.

Similar to the spectral method, an orthogonal, periodic wave function is used to guide the segmentation of the mesh into quadrangular domains. This time however, a standing wave on the mesh surface is optimised such that its iso-parameter lines are aligned with a conjugate input vector field defined by an orthogonal pair of unit vectors in the tangent plane of each vertex. In addition to the input vector field, a size field is used to guide wave lengths in the respective directions. The wave lengths of the function on the mesh are directly proportional to the edge lengths of the final quad mesh.

8.3.1 Wave construction

To generate a standing wave on the mesh surface, consider a scalar function χ defined on an input surface S

$$\chi(x,y,t) = f(x,y)\psi(t).$$

From the wave function definition, a constant a constant value wavelength $\lambda > 0$ can be derived

$$\frac{\nabla^2 f}{f} = \frac{\psi^{\cdot\cdot}}{c^2\psi} \equiv - \lambda.$$

Zhang et al. (2010) proceed to separate the equation into

$$f(x,y) = g(x)h(x)$$

to permit individual size control in either parameter direction, rewriting eq. 1 as

$$\frac{\partial^2 g}{\partial x^2} = -{\omega_x}^2 g \text{ and } \frac{\partial^2 h}{\partial y^2} = -{\omega_y}^2 y$$

With ω_x and ω_y satisfying ${\omega_x}^2 + {\omega_y}^2 = \lambda$, the two individual equations can be generally solved as

$$g(x) = A_x \cos(\omega_x x + c_x) \text{ and } h(y) = A_y \cos(\omega_y y + c_y)$$

where A_x, A_y, c_x and c_y are constants.

The functions inside the cosine terms are collected an represented as angle parameters $\theta = \omega_x x + c_x$ and $\varphi = \omega_y y + c_y$, yielding

$f(x,y) = \cos\theta * cos\varphi$

The supplied size fields μ, η is converted to the angular speeds ω_x and ω_y by the relationship $\omega_x = \frac{X}{\mu}$ and $\omega_y = \frac{Y}{\eta}$.

Like Ray et al. (2006), a first order Taylor approximation is used to establish a relation between the angle values and given mesh parameters. Recall
Figure 10 for the diagrammatic meaning of the symbols.

$$\theta j - \theta i = \omega_x X(e_{ij}) (p_j - p_i)$$

$$\phi j - \phi i = \omega_x X(e_{ij}) (p_j - p_i)$$

Setting $\alpha_{ij} = \theta_j - \theta_i$ and $\beta_{ij} = \phi_j - \phi_i$, the wave function can then be approximated as

$$F(p_i) = cos(\theta_i + \alpha_{ij}) * cos(\phi_i + \beta_{ij}) = c_{ij} * f_i$$

where

$$c_{ij} = \begin{bmatrix} cos\alpha_{ij} * cos\beta_{ij} \\ -cos\alpha_{ij} * sin\beta_{ij} \\ -sin\beta_{ij} * cos\beta_{ij} \\ sin\alpha_{ij} * sin\beta_{ij} \end{bmatrix} \text{ and } f_i = \begin{bmatrix} cos\theta_i * cos\phi_i \\ cos\theta_i * sin\phi_i \\ sin\theta_i * cos\phi_i \\ sin\theta_i * sin\phi_i \end{bmatrix}.$$

c_{ij} can directly deduced from the input mesh. f_i contains the unknowns to solve for.

Additional penalty terms, derived from trigonometric entities, ensure coherent, nonzero results:

$(cos\theta_i * cos\phi_i)^2 + (cos\theta_i * sin\phi_i)^2 + (sin\theta_i * cos\phi_i)^2 + (sin\theta_i * sin\phi_i)^2 = 1$

$(cos\theta_i * cos\phi_i) * (sin\theta_i * sin\phi_i) - (cos\theta_i * sin\phi_i) * (sin\theta_i * cos\phi_i) = 0.$

A nonlinear energy term is then minimised to find appropriate values for the mesh vertices

$$E(f_0, \dots f_n) = \sum_{e_{ij} \in E} \begin{pmatrix} (e_1 f_j - c_{ij} f_i)^2 \\ + (e_1 f_i - c_{ji} f_j)^2 \end{pmatrix} + w_1 \sum_{p_i \in P} (f_i f_j - 1)^2 + w_2 \sum_{p_i \in P} (f_i^T H f_i)^2 ,$$

where H = $\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$, e1 = $\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ and w_1 and w_2 are empirically found constants, set to 0.15 each.

The first row of the vector f_i contains the values which we substitute back into the original wave equations to attain wave values.

8.3.2 Feature constraints

Zhang et al. (2010) also propose a means of constraining the mesh edges to align with feature edges. As the sub-domain boundaries of the quadrangulation are aligned with the iso-parametric curves of the wave function, such boundary vertices must have a function value of

$sin\phi_i = 0 \; or \; sin\theta_i = 0.$

Similarly, corner vertices can be constrained to

$cos\theta_i * sin\phi_i = sin\theta_i * cos\phi_i = sin\theta_i * sin\phi_i = 0$

in the optimisation. A vertex is considered a corner vertex if its edge list contains multiple feature edges.

8.3.3 Optimisation

All relevant variables are extracted from the 3d-model and passed to Matlab, where the above objective function, a least squares system, is minimised iteratively using preconditioned conjugate gradients. Matlab permits the estimation of gradients using finite differences, but in practice this is too slow given the size of the problems. Gradient information is therefore provided explicitly in the form of a Jacobian matrix, which is an $m\ x\ n$ matrix recording values for the partial derivatives with respect to each unknown parameter in the equation. The Hessian matrix is estimated automatically by the solver. Jacobian and Hessian matrices are re-computed after each outer-loop iteration. Convergence is reasonable and computation times slightly slower but, at +/- 60s per 10k vertices, of the same order of magnitude as the ones achieved by Zhang et al. (2010).

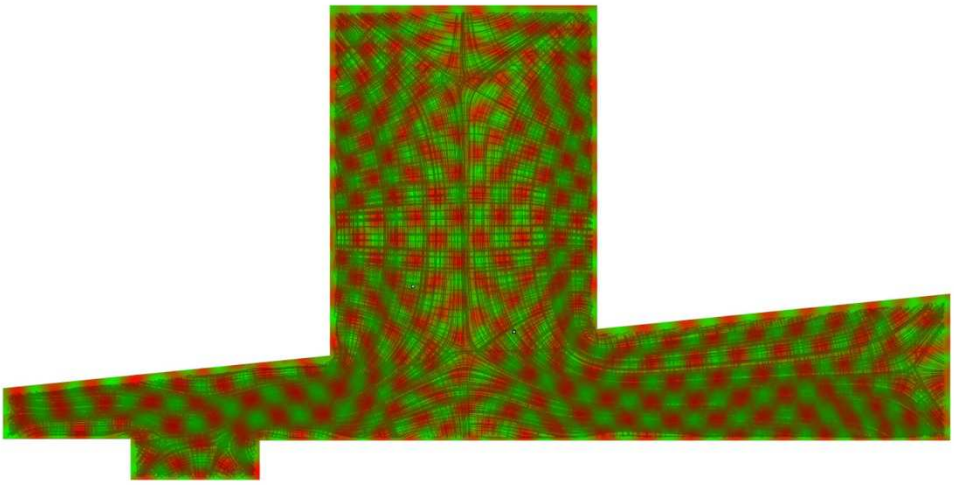

Figure 13 Wave function on glass goof model, with curvature field streamlines overlaid. The function converged nicely over the central section but the wavelength is too long to capture the additional detail in the bottom left corner.

8.3.4 Mesh extraction

The mesh is extracted via the creation of a Morse-Smale complex over the wave function, as has been done for the Laplacian eigenfunction in chapter 7.1. Charts are then parameterised using the method discussed in chapter 4.

Tests have shown that the Morse-Smale complex could be extracted more consistently on fine meshes and longer wave lengths as sampling errors occurred otherwise. Also, the charts were required to contain a minimum amount of triangular faces for the parameterisation to the quadrangular domain to be successful.

8.3.5 Convergence and results

The wave function formulation converged nicely over simple meshes, but errors appeared on more complicated shapes.

More testing needs to be done whether unreasonable size fields and bad weighing of the parameters caused poor convergence, or whether optimisation got trapped in local minima of the objective function.

Generating usable results involved plenty of parameter adjustment and significant simplification of the vector field. For the torus shape in Figure 14, the (known) size field was supplied manually and the wave converged perfectly.

Boundary constraints in particular affected convergence and satisfactory results were obtained again only on primitive models and when size fields were supplied rather than computed automatically.

Results were greatly improved by increasing the vertex count of the input mesh for more reliable sampling of the wave function, but this was penalised severely in terms of computation overhead.

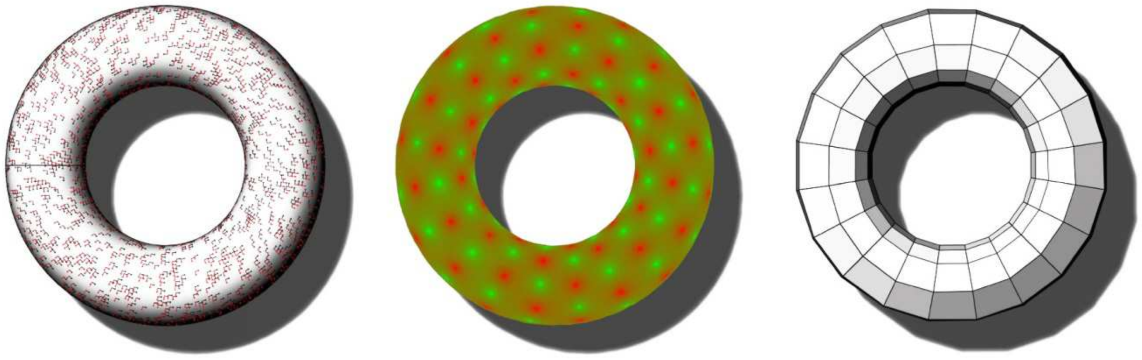

Figure 14 Torus remeshed using periodic wave function. Left: Curvature vector field displayed on input mesh of 12k vertices. The size field was supplied manually, depending on the distance to the centre of the torus. Middle: The wave function on the torus converged in 22s. Right: The extracted quad mesh (base complex). Vertices are located at maxima and minima of wave function.

9 Base complexes from constrained subdivision

The methods studied so far provide the means to approximate a given input geometry with all-quad meshes. When applying them to real-world case-studies the stringent mathematical formalism might prove restrictive to the freedom sought in design. It is not unlikely that the structuring of a surface is supposed to follow various local conditions or parameters. To formulate vector field and wave function to at once fulfil the multiplicity of biddings and constraints encountered in architectural design tasks is difficult at best.

A modelling pipeline based on subdivision surfaces has therefore been implemented to complement other methods. The idea of constructing a base complex to parameterise the input mesh remains, but here it is constructed manually. Given the frequently rather "simple" geometries in the built environment, is has ultimately been found that constructing a base complex manually has merits over its automation. For simple geometries, it can be more reliable, in particular when aided by vector field creation and visualisation techniques.

To comply with special constraints and considerations (see 5.1.1) likely to be found in architectural design, subdivision rules are augmented locally.

9.1 Subdivision surfaces

Subdivision surfaces are widely used in computer graphics to generate smooth surfaces from coarse base complexes. While in a standard tessellation, faces of the coarse mesh are successively subdivided without changing existing vertex positions, in surface subdivision, these locations are adjusted using specially defined weights along mesh edges.

Different weighing schemes have been proposed to achieve various criteria. The most widely used schemes are the Loop subdivision scheme for triangles and the Catmull-Clark scheme for quadrilateral meshes. In the limit, both schemes converge to surfaces which are continuous in terms of curvature everywhere except for at irregular vertices, where surfaces are continuous in terms of tangency.

Coherent surveys of popular subdivision methods and their applications can be found in (Zorin, 2006) and (Peter et al., 2009).

9.1.1 Subdivision rules

Subdivision rules are linear and defined only locally for mesh vertices. Commonly, new vertex positions are dependent only on the "stencil" surrounding a vertex. In case of the Loop and Catmull-Clark scheme, this stencil coincides with the one-ring around a vertex. After the first subdivision step, Catmull-Clark meshes are exclusively made up of quads. Figure 15 illustrates the standard weights applied to find vertex positions in said schemes. Special rules apply to boundary vertices.

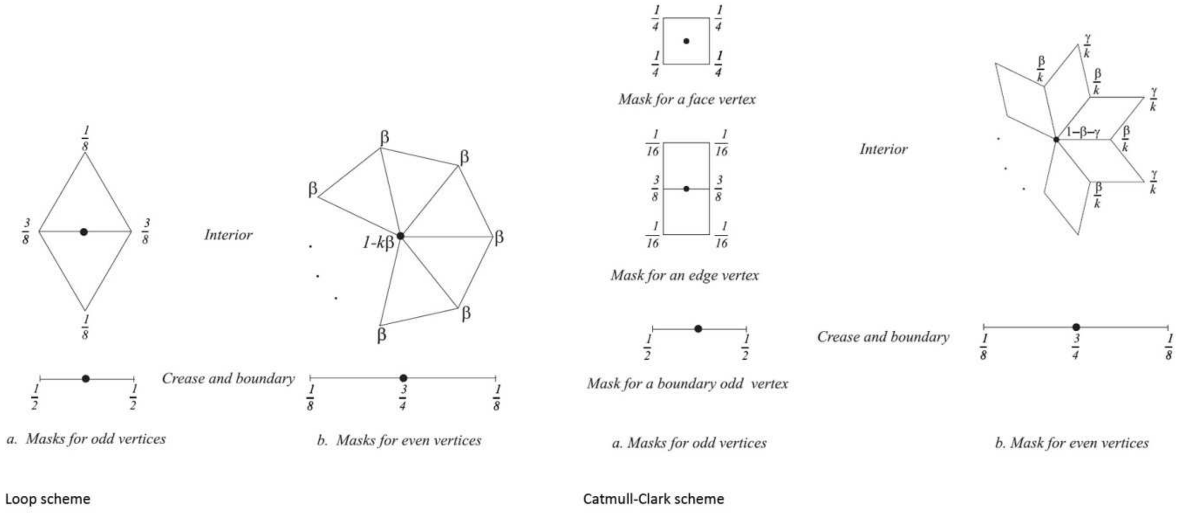

Figure 15 Subdivision rules for Loop(left) and Catmull-Clark (right) and schemes. Graphics courtesy of Schröder et al. (2009)For the Loop scheme, $\beta = \frac{3}{16}$ if the vertex valence val is 3, and $\beta = \frac{3}{8}val$ otherwise are used. For Catmull-Clark, $\beta = \frac{3}{2}val$ and $\gamma = \frac{1}{4}val$.

9.1.2 Limit surface

As mentioned above, meshes infinitely subdivided using the Loop or Catmull-Clark scheme converge to given surface which exhibit elegant geometric properties. One such property is that these surfaces are continuously differentiable which can be advantageous in many engineering applications.

The limit surface can be computed explicitly, without the need for computationally costly (infinite) subdivision steps using a technique put forward by Stam (1998). Correspondingly, vertices of coarser versions of the mesh can be pulled to the limit surface, giving a discrete sampling of the surface. Once the desired edge lengths are achieved, the mesh vertices can be projected onto the limit surface using

$$v_i^{\infty} = \frac{val^2 v_i + \sum_{j=\in V_i} 4e_{ij} + f_{ij}}{val(val+5)},$$

where val denotes the valence of vertex v_i (Sharp, 2000).

9.2 Constrained subdivision

In the implemented software, standard rules have been adapted to conform to particular boundary conditions present in architectural design.

9.2.1 Boundary and crease curves

The definition of an exact perimeter curve is quintessential to architectural design. By supplying perimeter curves as additional input to the subdivision process, local boundary conditions are overridden to match the new outline.

When a boundary curve touching two adjacent face vertices is detected, the newly created “edge point” is snapped to the halfway point of the relevant curve segment. Care must be taken that here the NURBS-parameterisation cannot be used as these parameters are generally not linear.

Similarly, it may be desirable to have a series of interior vertices to align with specific curves. In architecture such requirement may arise due to required alignment with supporting structure. Where in computer graphics weights are designed to still generate C2-continuous creases due to defects when rendering, kinks are usually expressed in architecture. Similar to the definition of boundary curves, crease curves can be defined which are later taken up by the subdivided mesh. Boundary constraints in action are shown in Figure 16.

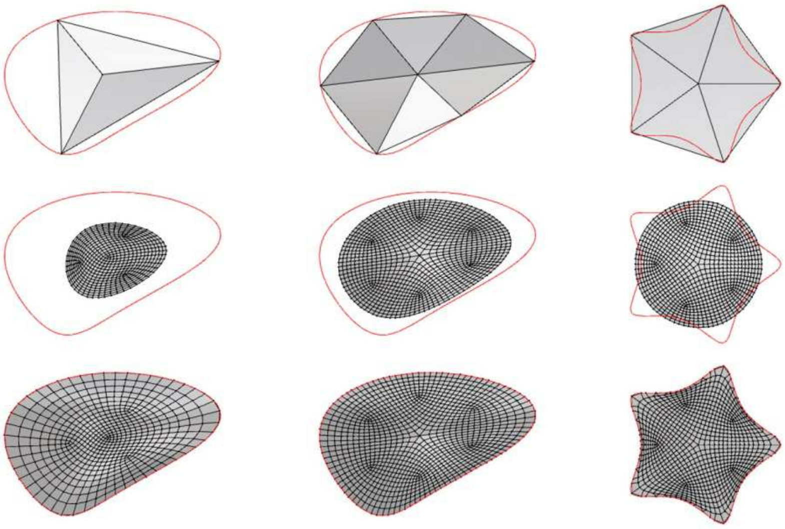

Figure 16 Top: problem definition with boundary curve and base complex. Middle: Subdivision without boundary constraint. Bottom: Constrained subdivision. Number and position of irregular vertices depends on the base complex

9.2.2 Boundary surfaces

In addition to the special rules for crease curves, rules for meeting boundary surfaces tangentially are specified. In case the final mesh is not projected onto the limit surface, boundary vertices and their neighbours are simply locked in place at each iteration step. When working with limit surfaces, weights described by Levin (1999) are used to generate Neumann conditions for the limit surface, see Figure 17.

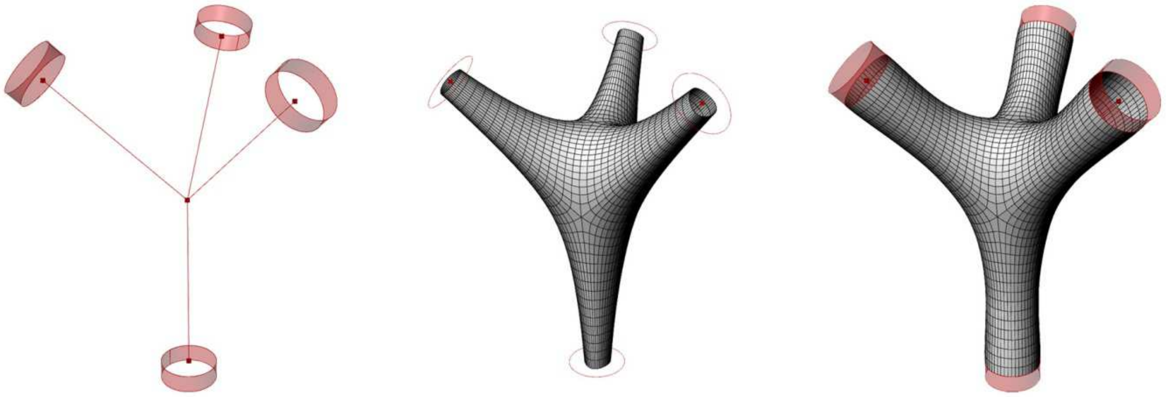

Figure 17 Branching node design using subdivision. Left: topological setup. Centre: Subdivision with standard Catmull-Clark boundary rules. Right: Subdivision with tangential constraints imposed by the cylindrical surfaces shown in red.

9.2.3 Reverse subdivision

The implemented software also permits the inversion of every subdivision step, a property that could later be used for structural optimisation (see chapter 12.3). Lanquetin et Neveu (Lanquetin and Neveu, 2009) make use of the linearity of the subdivision process to compute vertex positions of a coarse mesh. Note that these steps only return fully reversible results for meshes which are already the result of subdivision (Yacine et al., 2010).

9.2.4 Remeshing using subdivision

For the purpose of remeshing, a coarse mesh is constructed from vertices of a given input mesh. This is used as basis for subdivision. Rather than using the scheme-specific weights, the vertices of the newly subdivided mesh are placed onto the input mesh using the parameterisation method described in chapter 4, overriding the limit surface of the coarse mesh. Boundary constraints are applied to ensure the boundaries of input- and output mesh remain aligned.

10 Subsequent optimisation

So far the concern of this thesis lay with the creation of aesthetic grid layouts on surfaces. As discussed in chapter 1.1, many architectural applications are equally concerned with covering the surface using panels. Specific algorithms can refine the meshes generated with the methods outlined in the previous chapters in several post-processing steps.

One of the main advantages of working with meshes is that many optimisation techniques for efficient panelling can be formulated sharing the same finite elements, sparing tedious conversion between workflows. Unlike remeshing, which aims at approximating an existing surface or mesh, the optimisation steps can affect the input geometry significantly whilst generally leaving the topology unchanged.

10.1 Multi-objective optimisation

To provide a design tool satisfying the need for intuitive user control, a dynamic relaxation suite for Grasshopper was developed.

Dynamic relaxation is a process whereby nodes of a discretised topological system are made to oscillate about an equilibrium position (Barnes et al., 1977). This position, determined by stiffness terms defined in between nodes as well as global forces, is reached in an iterative dynamic process. The topological relationships already defined in the meshes make it well suited for problem definitions.

A modular build-up allows an easy and multi-objective problem definition. Specific behaviours are assigned to input geometry either referenced from Rhinoceros 3d or generated directly in Grasshopper using individual components. These behaviours assign forces which for example turn mesh edges into springs or impose certain optimisation constraints on vertices of an input mesh.

The program gives on-the-fly visual feedback on interactive parameter changes by the user, such as force magnitudes or node relocation, permitting instant evaluation of the results.

For completeness, the dynamic relaxation tool can also be used as a basic form-finding tool for the creation of hanging cloth models.

Planarisation

A frequent requirement for building envelopes is their construction from planar elements. In practice, arbitrary input meshes cannot be optimised toward planarity without significant distortion (in particular, quads collapsing to triangles or lines) or significant change in boundary conditions. Meshes exhibiting low initial warp, such as meshes following principal curvature lines, can however be further optimised toward near-planarity. Lax construction tolerances or flexibility in the material often permit deviation from planarity in favour of a more anisotropic layout.

For planarization, the distance between the diagonals is minimised for every quad face. When working with subdivision, the planarisation constraint is applied subsequently to every subdivision step (Pottmann et al., 2007).

Constrained force equalisation

Dynamic relaxation can be used to define spring or distance constraint forces along mesh edges to equalise residual forces within the members. This is similar to Laplacian smoothing which would eventually lead to an approximation of a minimal surface if all spring target lengths are set to zero. When remeshing or for surface drape, vertex locations can be restricted to guide surfaces, meshes or planes.

When using quadrangular meshes, such spring forces can be set up to act along quad diagonals to simulate a primitive shear force.

Conical meshes

"Conical" meshes have the beautiful property of constant face offsets everywhere (Liu et al., 2006). This is particularly interesting to architecture as multi-layer envelopes are the most dominant form of construction of our time. Optimising for the conical property requires the opposing angles of a valence four vertex to be equal, Again, this problem can be geometrically imposed on vertices with valence 4 by the dynamic relaxation.

The in-built mesh offset function in Rhinocommon does not permit constant face offsets; a basic bespoke method has therefore been implemented into Grasshopper.

Curvature operators

Since curvature information can locally be computed on the fly for every vertex of the mesh (see Appendix A), it is possible to set up forces acting along mean or principal curvature vectors. An example of such a force is diffusion flow, whereby minimal surfaces can be achieved by locally trying to reduce curvature. Using a similar setup, the curvature variation from one vertex to its neighbours can be minimised.

Bending

Bending can be simulated be attempting to increase the cross product between two opposing edges emanating from an even-valence vertex. This way of simulating bending actually also provides an interesting tool for mesh smoothing. Laplacian smoothing, which places vertices at the barycentres of their neighbours can produce strong kinks, particularly if a quad mesh contains triangles near boundaries. This can be alleviated by using "bending" instead of classic Laplacian smoothing, or a combined term.

11 Case study

The performance of the software tools was assessed by their contribution to the design of an actual building project. The case study, a steel and glass roof in Shanghai, was provided by Schlaich Bergermann and Partners.

11.1 Glass roof, Shanghai, China

The glass roof is part of a shopping mall project in Shanghai's Luwan district, due to open in 2013. The winning competition entry indicated three main aisles merging tangentially with a large central atrium over four levels of retail.

In an early design meeting it was decided to cover the internal routes with a continuous glass roof, visually suggesting fluidity. As indicated on an early sketch shown in Figure 18, the roof was to bow down and touch the ground, creating a vortex shaped feature rather than spanning the 52m diameter of the circular atrium uninterruptedly. The weight of the roof would then be distributed between an edge beam around the atrium perimeter and the doubly-curved geometry of the central funnel.

Given the real nature of the project, not only were the eventually generated geometries of importance in the evaluation, but just as crucial was their integration in the project workflow which involved coordination with architects, structural and façade engineers. Equally, the ability of the respective techniques to respond to sudden local or global design changes was crucial. Figure 19 gives an overview of the individual design an optimisation steps performed.

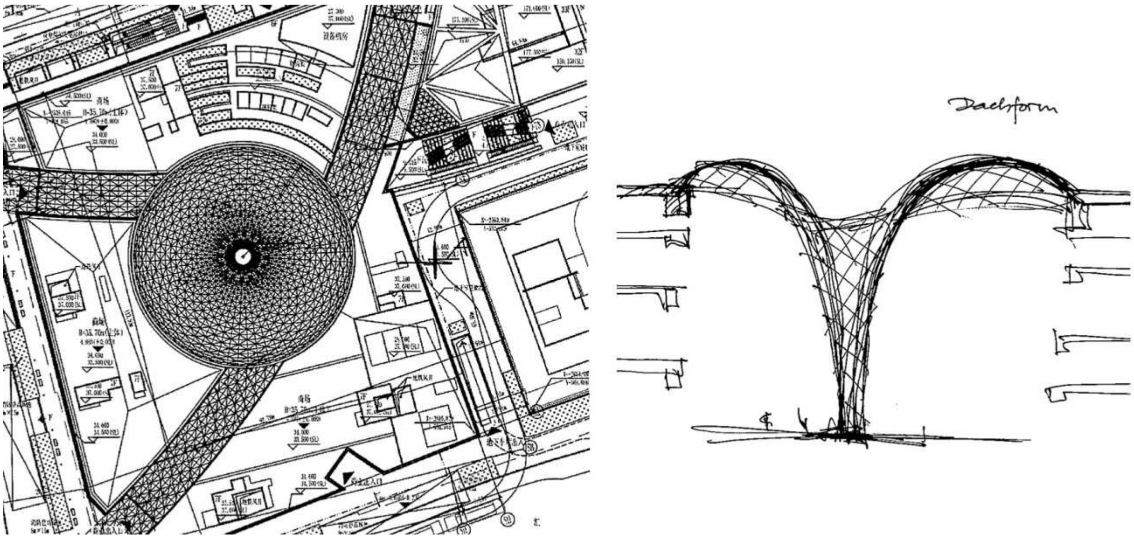

Figure 18 Initial project sketches. Left: The roof level plan. Right: Sketch of the funnel shape in section. Images courtesy of Schlaich Bergermann and Partners

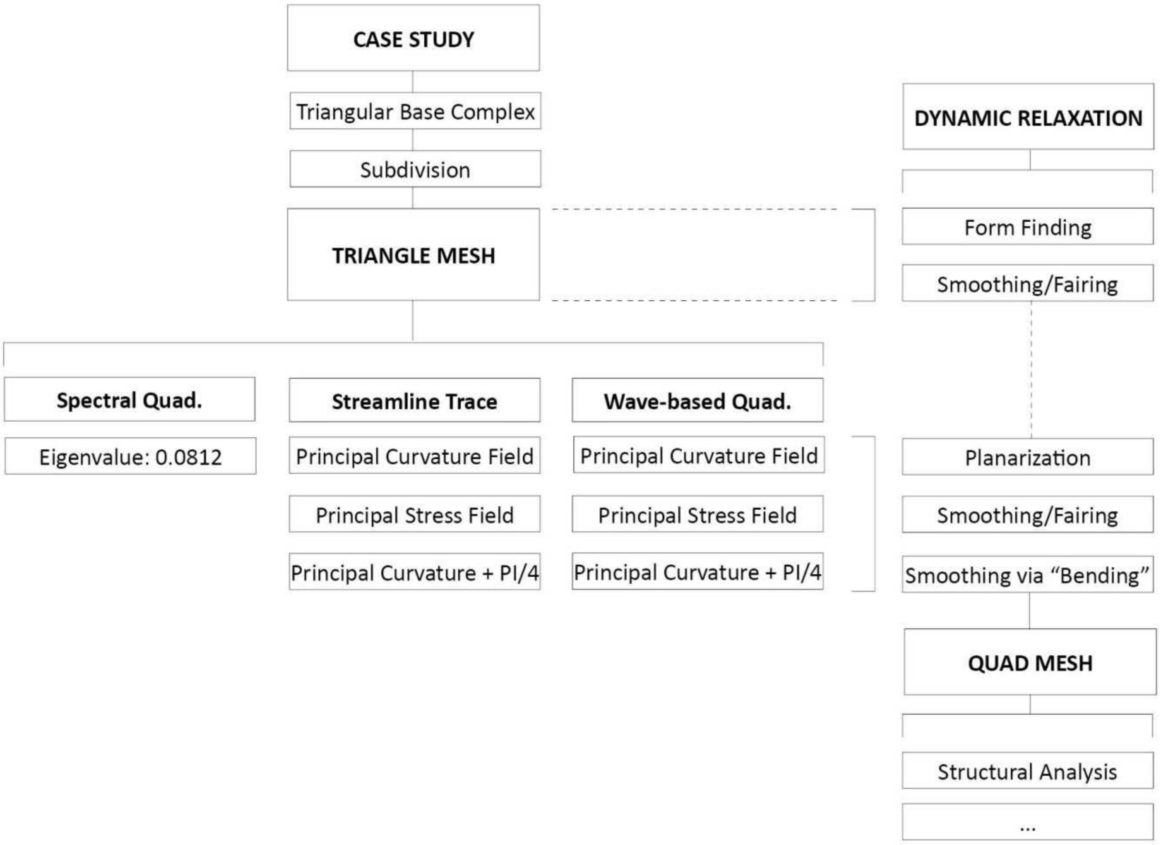

Figure 19 Diagrammatic overview of case study design and optimisation steps

11.2 Form finding and triangular mesh

11.2.1 Triangle mesh from subdivision

Subdivision was the obvious choice to generate an initial structure. Based on the original intention, a base complex was modelled. The central atrium was covered with six triangles, arranged such that nodes were in strategic positions. Subdivided once, and with outlying faces omitted this generated a larger truncated triangle. The corridors, themselves simple chains of triangles would then connect to the edges where the initial shape was cut.

The funnel was generated by discarding the central roof triangle and instead adding additional triangles arranged vertically until a circular boundary curve on ground level was reached. This initial complex was then subdivided until the desired edge length (1.5m – 2.0m) was best approximated. This process is shown in Figure 20. Subdivision was constrained such that new boundary vertices would snap to the top and bottom perimeter curve.

This simple topology of the base complex would ensure that the entire mesh would contain only three irregular vertices of valence seven, distributed evenly around the rotational axis of the atrium and at the apex of the transition curve from vertical to horizontal roof. Furthermore, by creating initial triangles of similar sizes, variation in the final edge lengths was expected to be minimal.

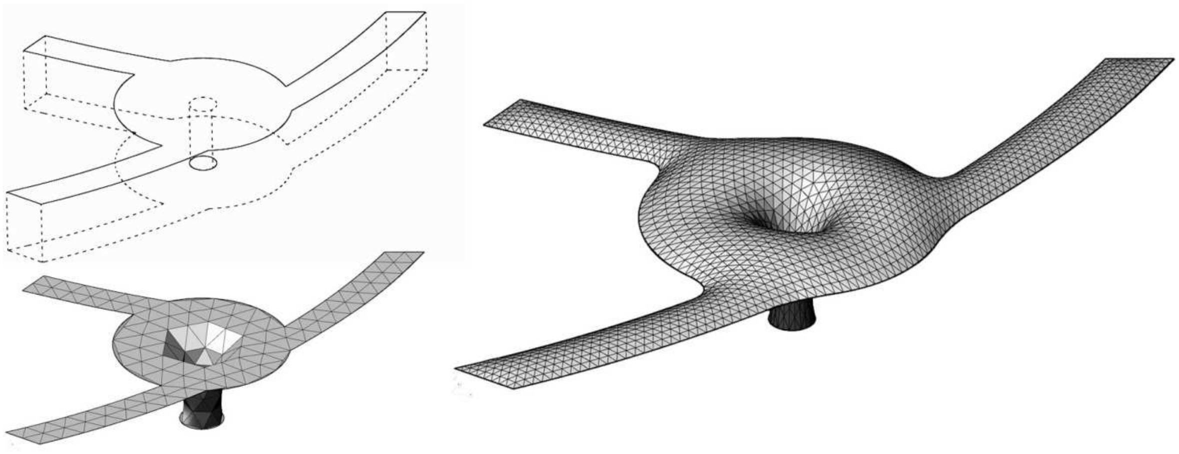

Figure 20 Progression from base complex to final triangle mesh. Top left: Diagrammatic illustration of perimeter curves. Bottom left: Simple triangular base complex. The complex contains only three irregular vertices whilst approximating the perimeter with mostly isotropic faces. Right: The subdivided mesh after initial form finding using dynamic relaxation.

11.2.2 Dynamic relaxation

By fixing boundary vertices in space and applying a zero-length spring to each mesh edge, an inverted gravitational force was counteracted in the fashion of a hanging net model. Spring strengths were adjusted on the fly until a reasonable span to depth ratio was achieved.

This first attempt already produced a convincing result, except for a visually distracting convergence of the edges to the central points of the atrium arc segments. In a second iteration, triangles outlying of the polylines directly connecting the corridors were therefore omitted from the mesh, and the new boundary vertices pulled to the perimeter curves, where springs would distribute them evenly. Subsequent relaxation led to an increased distortion of the triangles, but the topology was felt to be more naturally responsive of the plan shape. The enlarged variation in edge lengths was remedied by Laplacian smoothing.

As the design progressed, several slight alterations to the perimeter curve were made, the most visible of which was the filleting of the corners where the aisles met the central atrium. The parametric set-up required only the automated pulling of the corner vertices to the adjusted boundary curve the relaxation procedure could be performed once again. This configuration permitted the retention of downstream relationships such as the node numbering in the finite element package used to perform the structural analysis.

11.2.3 Quad-mesh generation

Quadrangular alternatives to the triangular panels were studied. The triangle mesh would serve as the input geometry to the quadrangulation.

To evaluate the suitability of the meshing techniques, base complexes were generated using spectral methods, manually with the aid of streamline visualisation, as well as wave-based methods. Spectral quadrangulation did not require the computation of a vector field. For streamline trace and wave-based quadrangulation, meshes following principal curvature and principal stress lines were computed. A third, alternative, base complex was constructed manually.

To ensure vector field (and later wave function) would be sufficiently sampled, the triangle mesh was subdivided until edge lengths were on average one tenth of the desired steel member length of the roof, leading to approximately 50,000 vertices.

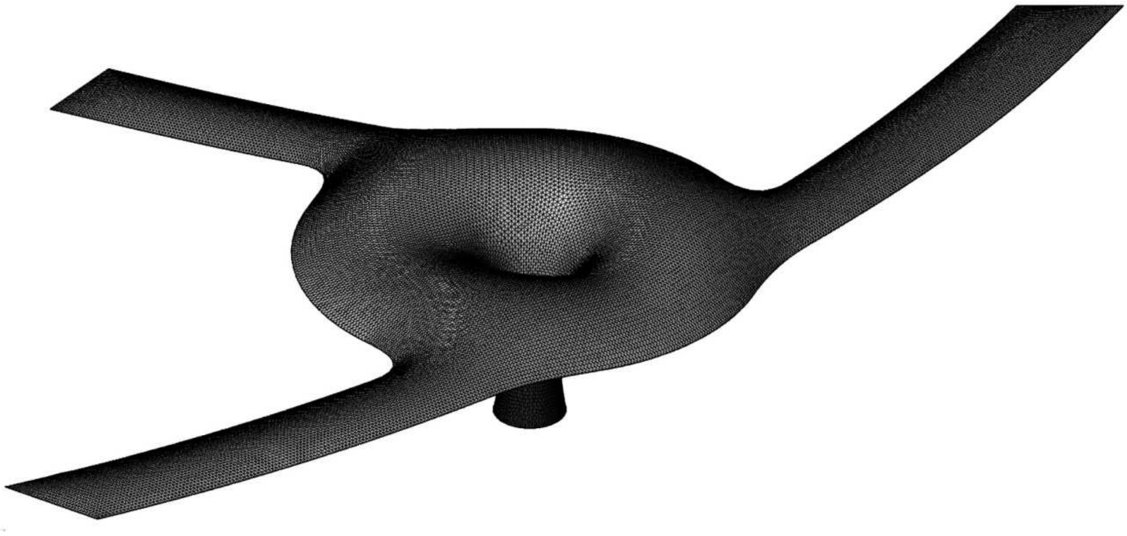

Figure 21 The input mesh of 50,000 vertices was generated by twice subdividing the triangular mesh

11.3 Benchmark: Graphite

To evaluate the performance of the proposed tools against results achievable using other methods, the model was first imported into INRIA's open-source mesh processing package Graphite. Graphite has been shown to produce outstanding quad meshing results for computer graphics applications but is yet to be used seriously on building projects.

Following vector field generation and pre-processing, it was found that Graphite gave little control over location and type of singularity. Albeit nice isotropic quads were produced, singularities were frequently encountered close to another, even when using a highly-smoothed vector field.

At the time of writing, locking the parameter value of border vertices would make Graphite return an error, which meant that jagged boundaries could not be avoided. Also, the program returned a closed mesh, which implied the requirement to delete a large number of faces manually. Graphite did however use a linear method to generate a parameterisation, which meant that result could be obtained within seconds, far superior to any of the automated solutions chosen in this work, which took at least 3 and 4 minutes to converge. The quick results from Graphite did provided useful guidance to construct a manual base complex from guide curves.

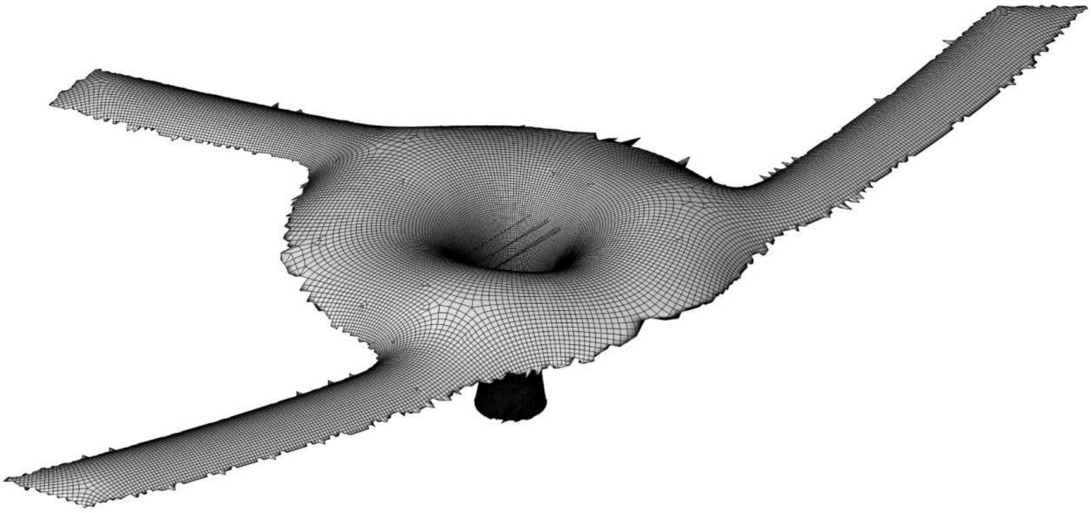

Figure 22 Quad mesh from Graphite's periodic global parameterisation

Once a benchmark was set (and found in many ways to be inadequate), the tools developed by the author were put to the test.

11.4 Spectral quad mesh

Since it is vector field independent, spectral quadrangulation was first attempted. Following the procedure outlined in chapter 6.1.2, sampled eigenfunctions were extracted from the cotangent-weight Laplacian matrix. Computation of a single eigenvector took less than a second. Vectors of appropriate wave length were found quickly, but as there are as many eigenvectors as there a vertices (stipulated by the size of the Laplacian matrix), there were still hundreds of eigenfunctions to choose from. For the meshes shown below, 100 eigenvectors were computed at once. As the Morse-Smale complex took less than a second to compute and display a quad mesh for a given eigenvector, it was possible to manually cycle through the options and select an appropriate one. After a few minutes a nicely distributed function of correct wavelength was found, but when remeshing, mesh quality along boundaries remained poor.

Almost all meshes still exhibited defects which needed to be corrected. This was done in parts by effecting post-processing steps when computing the Morse-Smale complex, but manual adjustments were still required. The converging geometry of the funnel in particular was susceptible to poor results, as were the ends of the corridors.

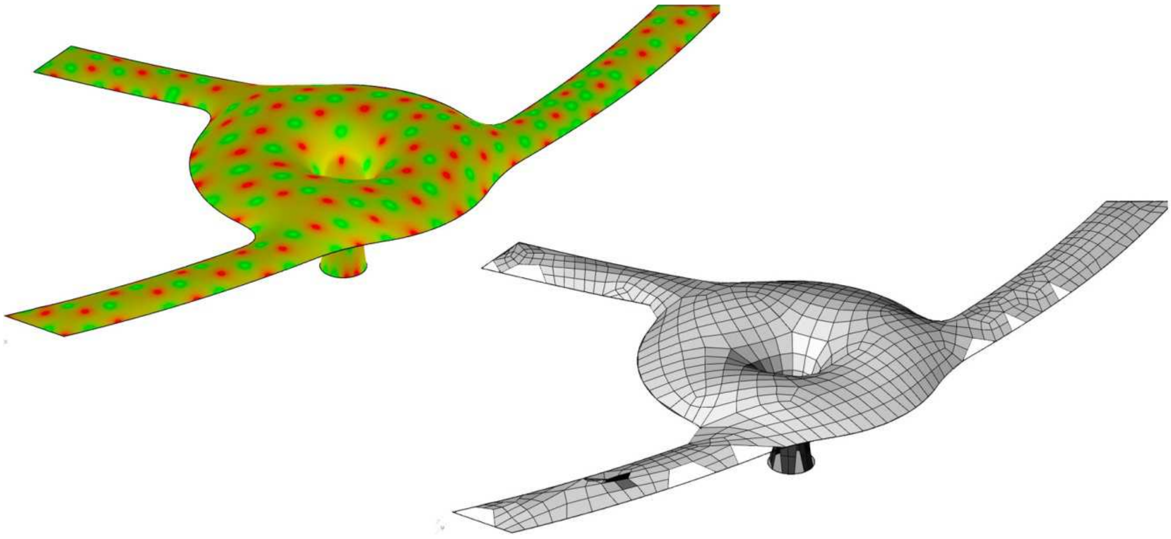

Figure 23 Laplacian eigenfunction (left, eigenvalue = 0.020183) and resulting quad mesh (right)

11.5 Meshes from principal curvature field

To give control over orientation of edges and locations of singularities, the mesh was embedded with vector field data. First, the principal curvature field was chosen. Curvatures and principal curvature directions were found using the method described in Appendix A. Despite multiple subdivision steps ensuring a smooth mesh in theory, the estimates of the curvature directions in isolated regions appeared affected by sampling errors, possibly due to the curvature approaching zero in certain areas. For this reason, the vector field had first to be smoothed. The resulting mesh is shown in Figure 24.

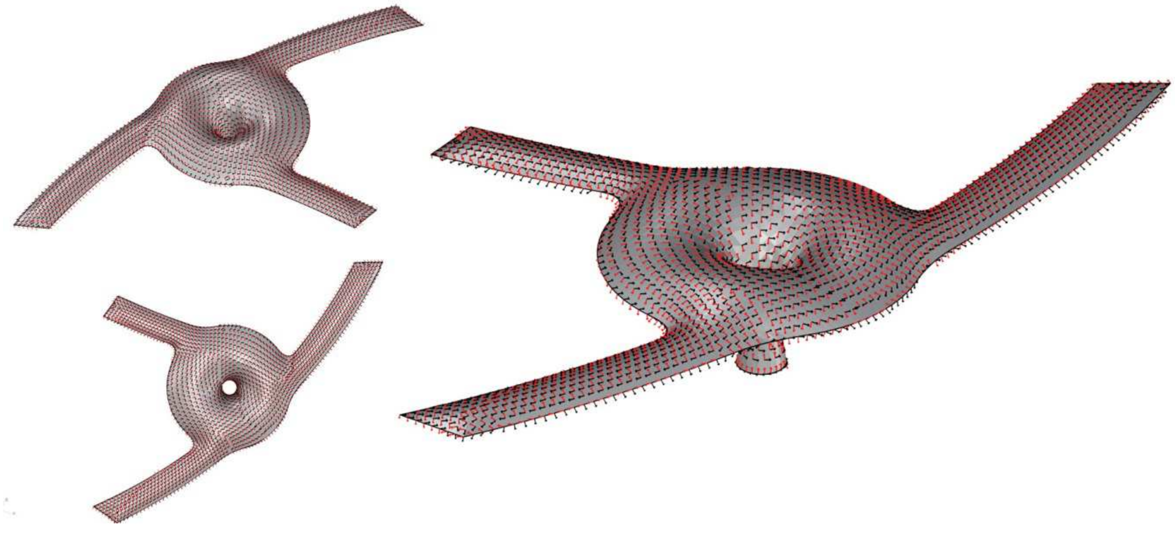

Figure 24 Maximum (red) and minimum (black) curvature field vectors displayed on the input mesh

11.5.1 Streamline trace

To generate a quadrangular base complex, the vector field was visualised by tracing streamline starting from every mesh vertex. Singularities of the vector field were also computed. The curvature field returned three singularities of valence three at the halfway points between the corridor “mouths” and the transition to the funnel. These locations coincided with the highest points of the roof.

Streamlines were traced for the asymptotic directions of the vector field associated with each singularity. This procedure revealed a comprehensible topological structure which was in fact simple enough for a very regular base complex to be generated from manually drawn mesh quads in only a matter of minutes. The remeshing result is shown in Figure 25. Only at the end of the corridors are the perimeter curves not parallel with the curvature field, resulting in predominantly quadrangular elements, even along the perimeter.

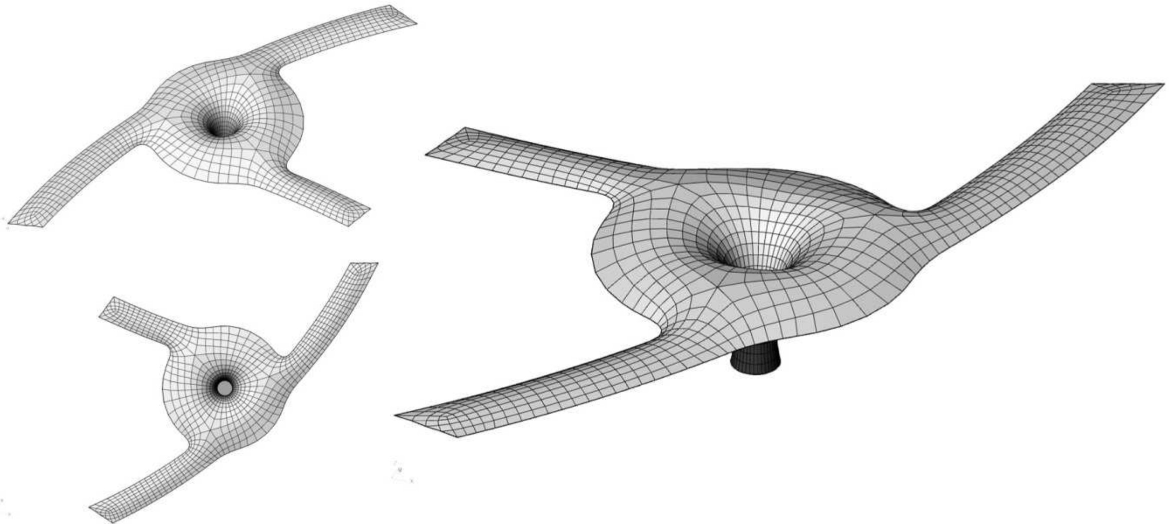

Figure 25 Mesh following principal curvature lines.

11.5.2 Wave-based quadrangulation

The wave-based quadrangulation method was also put to the test once curvature information was in place.

Size field pre-processing

When supplied with a constant size field of 2m lengths prior to pre-processing, the meshing algorithm would have to automatically place singularities and locally deviate from the vector field to meet isotropy requirements. A constant-length size field of 2m was supplied. This top-down size field produced several areas where rapid vibration was encountered. It was therefore subjected to the pre-processing steps outlined in chapter 8.1.4 to reduce curl and the occurrence of rapid vibration.

Wave length and convergence

The alignment of the wave function was studied for parametrically varied wave lengths of 2.63m, 3.5m and 5.25m as these lengths would evenly divide the corridors in transverse direction.

The supplied size field seemed to prohibit a globally successful convergence of the optimisation algorithm. A regular, field aligned wave function emerged over most of the surface, but for almost all cases there remained deficiencies which disallowed successful mesh creation everywhere. The solver converged more quickly the larger the value, but only the smaller values captured enough detail for successful mesh creation later on. Convergence was slower and even less reliable for meshes where vertices had been assigned specific parameter values to line up wave function and mesh boundary.

Since convergence appeared to get trapped in local minima, a less elegant post-processing step was included in the mesh extraction process. Boundary vertices of the quad-mesh were pulled to the closest point on the perimeter curve.

Figure 26 suggests that the base complex could not match the quality of those created manually, which is why in on subsequent projects the latter method has taken precedence.

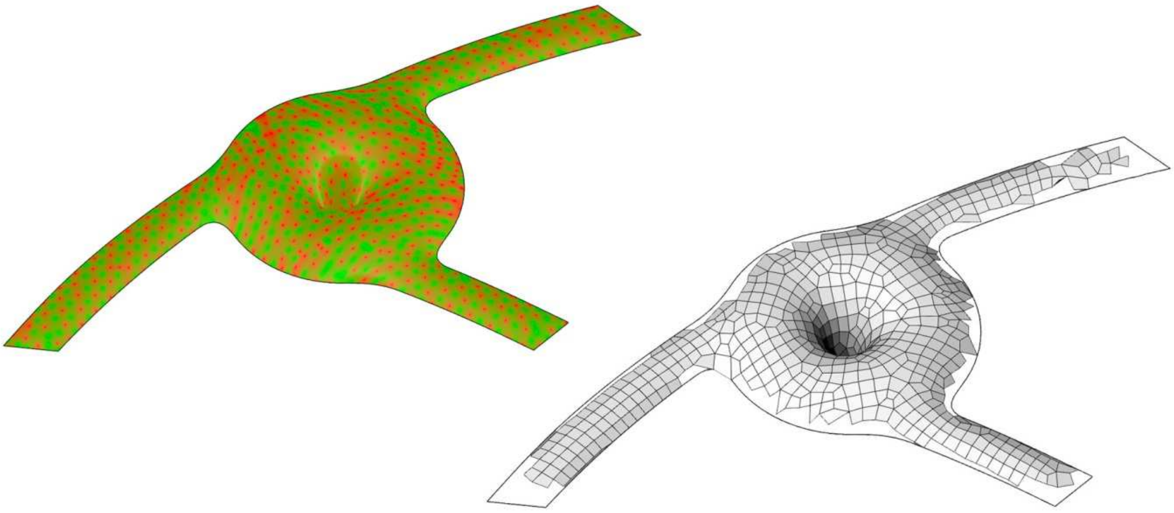

Figure 26 Wave based quadrangulation. Left: The wave function on the input mesh, wave length = 2.63m. The optimisation converged poorly in areas of clusterered maxima/minima. Right: The unedited quadrangulation result. The quad edges generally follow the curvature lines but quadrangulation failed near boundaries.

11.6 Mesh from principal stress field

As has been shown, the software permits the alignment of the mesh edges with any given conjugate vector field. One such field is the principal stress field. Envelopes where elements are aligned with the principal stress field have been shown to require reduced cross-sections in members for certain load cases (Schiftner and Balzer, 2010). This effect is reduced due to the multiplicity of loads expected on a structure over its lifetime. Stresses computed using a simplified material model would be sufficient to attain a qualitative impression of the topology of the principal stress and the directions of the stress field. A simulation was set-up in Millipede (Sawapan, 2012), a finite element analysis plug-in to Grasshopper to estimate stress directions. Edges were tentatively assigned the properties of 100mm diameter steel tubes, and deformation studied under dead load.

Though not set up for physical accuracy, the deformation of the shell seemed to deform reasonably in qualitative terms, as shown exaggeratedly in Figure 27.

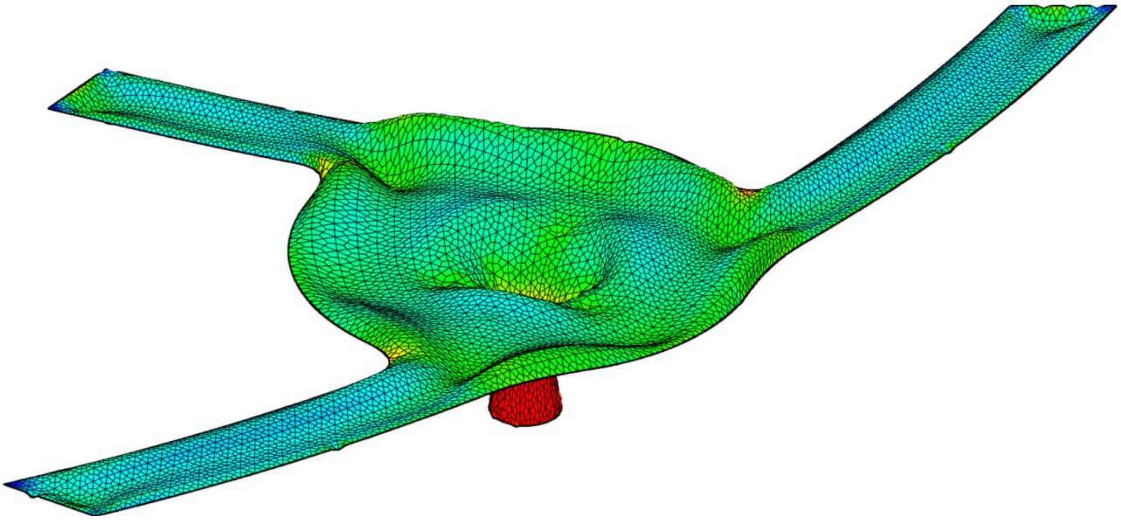

Figure 27 Exaggerated deformation of mesh under dead load only. The resulting stress field was used to guide remeshing.

Principal stress direction could not be extracted explicitly. Instead Millipede was used to export polyline representations of the streamlines, which were rebuilt as NURBS-curves and used as input to generate a conjugate vector field as described in chapter 8.1.3.

The streamlines were somewhat degenerate at the transition from the vertical to the horizontal parts of the funnel but smoothing the vector field reduced this to only three rotationally symmetric singularities.

Again, with streamlines and singularities displayed, constructing a base complex manually took only a matter of minutes. Some streamlines of the field are displayed in Figure 28, the corresponding mesh is illustrated in Figure 29.

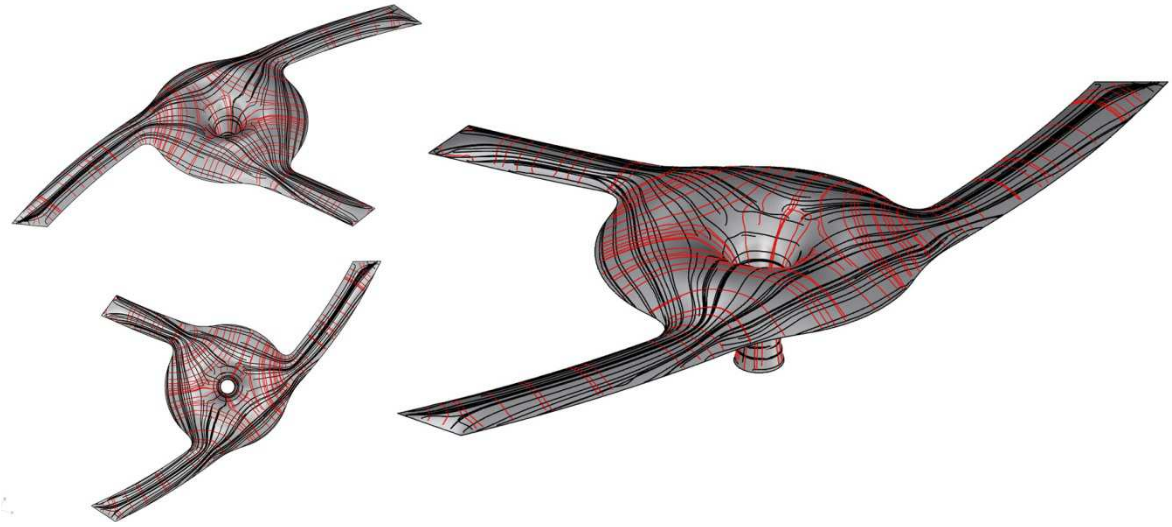

Figure 28 Maximum (red) and minimum (black) principal stress lines displayed on input surface

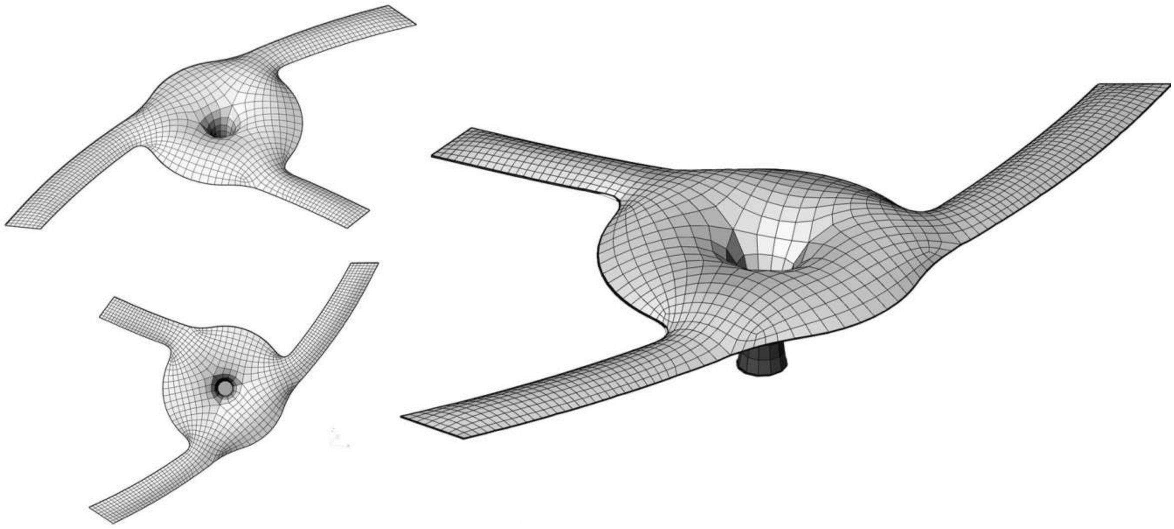

Figure 29 Quad mesh from principal stress field under dead load

11.6.1 Custom base complex

In addition to the curvature and vector field, a third alternative was studied based on a manually constructed base complex. The base complex was created in such manner as to move the three singularities encountered in the curvature field from the top of the roof over to the transition point between funnel and atrium roof. This resulted in only three singularities radially distributed around the funnel, with the additional advantage of them being only of valence 5. Also, edge lengths could be less diverging in this option as the strategic placement of the singularities lead to a smaller number of quads that needed to surround the vertical part of the funnel.

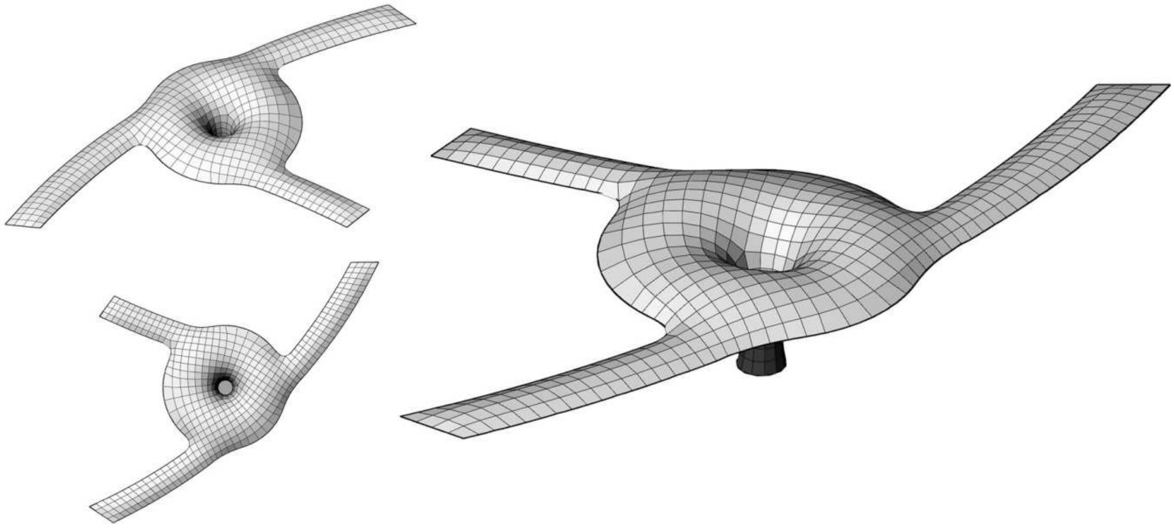

Figure 30 Mesh from custom base complex

11.7 Subsequent optimisation

11.7.1 Smoothing

To smooth the meshes, a combination of several components of the dynamic relaxation suite was used.

An even distribution of boundary vertices along the perimeter curves was desired. For this reason, boundary vertices were constrained to the perimeter curve while springs forces defined only along boundary edges were equalised. Once the boundary vertices were in the desired position, a combination of shear forces inside the quad panels, contraction forces along edges and bending forces which align opposing vertex edges were combined into a joint solution with user-adjustable weights. While adjustments were made, the vertices were constrained to the original triangle mesh such that the original shape was sufficiently approximated.

11.7.2 Planarisation of panels

Further to the topological definition and implications on structural performance, the suitability of the mesh layouts for planar panelling was studied.

To estimate the degree of deviation from planarity, the distance between the diagonals to the shortest edge length of each quadrangular panel were related for the three meshes. For comparison, all results of the optimisation shown in the next section use the same colour scheme. In a gradient going from green over yellow to red, green panels indicate zero warp, whilst red panels exhibit a ratio of 5% warp or more.

Since single-sheet glass was to be used as cladding material, a deviation of up to five per cent was believed to be admissible before specialist material would be required. Inside this tolerance, the glass would not have to be bent in advance but rather be delivered to site as flat panels and pulled into shape by clamps mounted on the steel frame.

Warp was measured and displayed using another custom component in Grasshopper.

The mesh geometry was then optimised using dynamic relaxation (see 10.1). The bespoke solver was set up to increase planarity while penalising deviation from the input geometry (by assuming a spring between the initial vertex location and the new position at each iteration.

Results

As expected, the panels following the principal curvature lines were by far the most suitable for planarization. In fact, all panels could be made flat to within a tolerance of 0.1% while keeping deviation from the initial geometry at a minimum. A colour-coded impression of the original mesh is shown in Figure 31. The greatly improved mesh from planar quadrilaterals can be seen in Figure 32. The small defects at the transition points between corridors and atrium were later adjusted manually.

Similar quality results could not be achieved for the principal stress or the custom mesh. The optimisation procedure caused edges to collapse, in effect creating triangles at certain (irregular) positions of the mesh. It also caused the geometry to crease, which would have great visual and structural penalties. Figure 33 shows that prior to optimising; both the mesh from principal stress field and the manually created one exhibited much greater warp than the principal curvature field. As shown in Figure 34, following the optimisation, panels were in parts highly anisotropic.

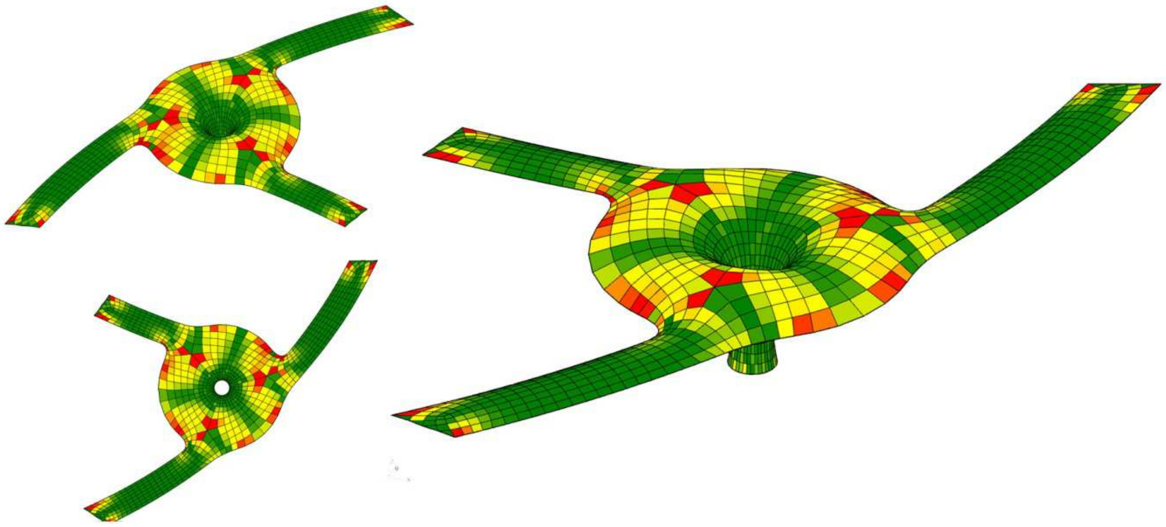

Figure 31 Planarity of panels pre-optimisation for the curvature field. Panels in the corridors are mostly flat but simply constraining the mesh to the initial triangle mesh geometry caused warp greater than 5% for interior faces. Still, this was considerably less than for other fields

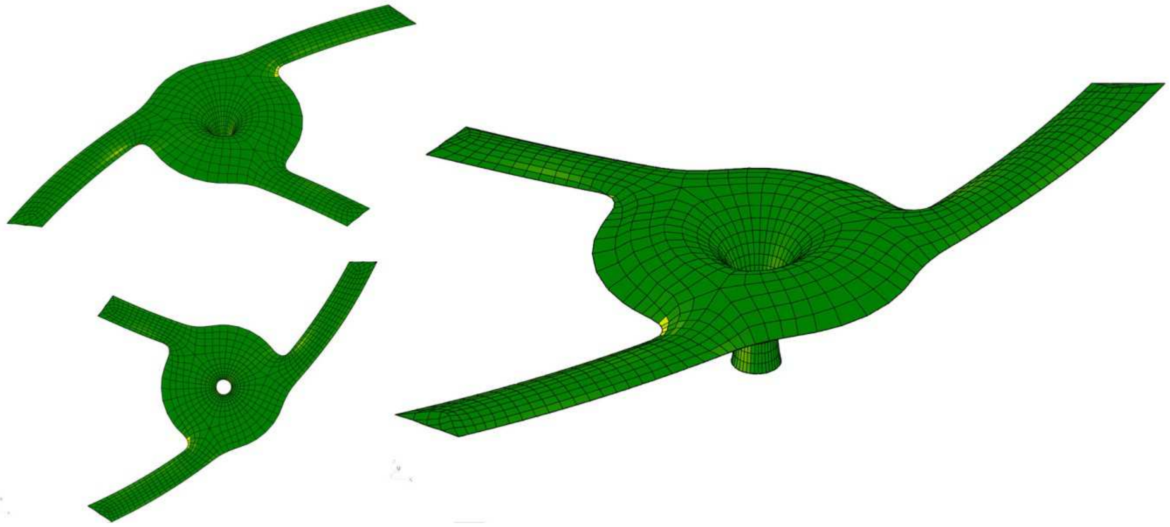

Figure 32 Planarity of panels post-optimisation. Nearly all panels are flat. Only small errors occurred at the interface between corridors and atrium, which had to do with fixing the boundary vertices while optimising. This was fixed in a subsequent step.

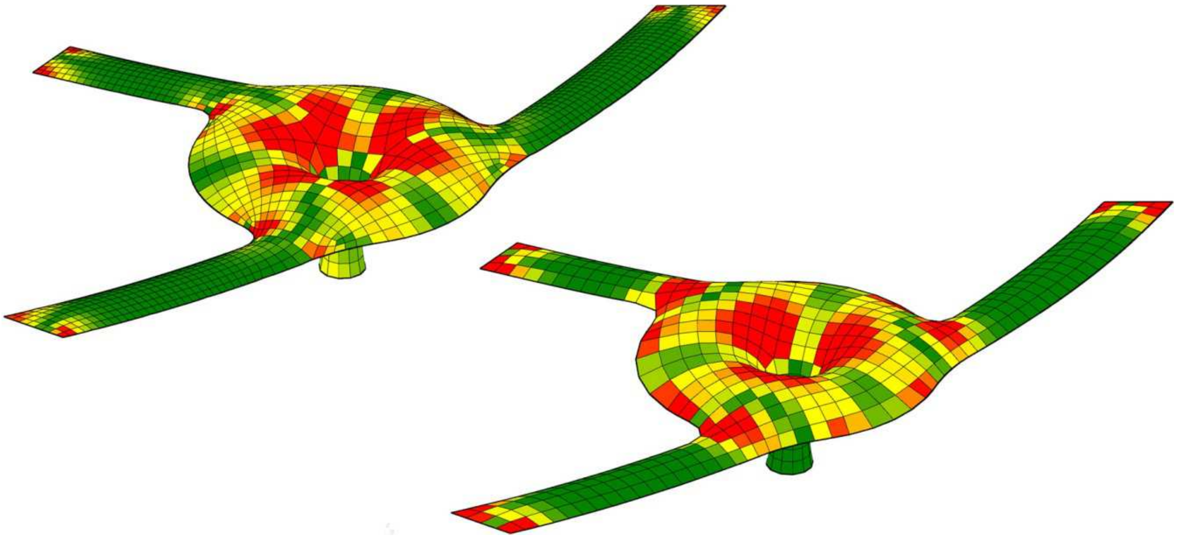

Figure 33 Comparison of deviation from planarity for different panel layouts, prior to optimisation. Left: Mesh from principal stress field. Right: Mesh from custom base complex

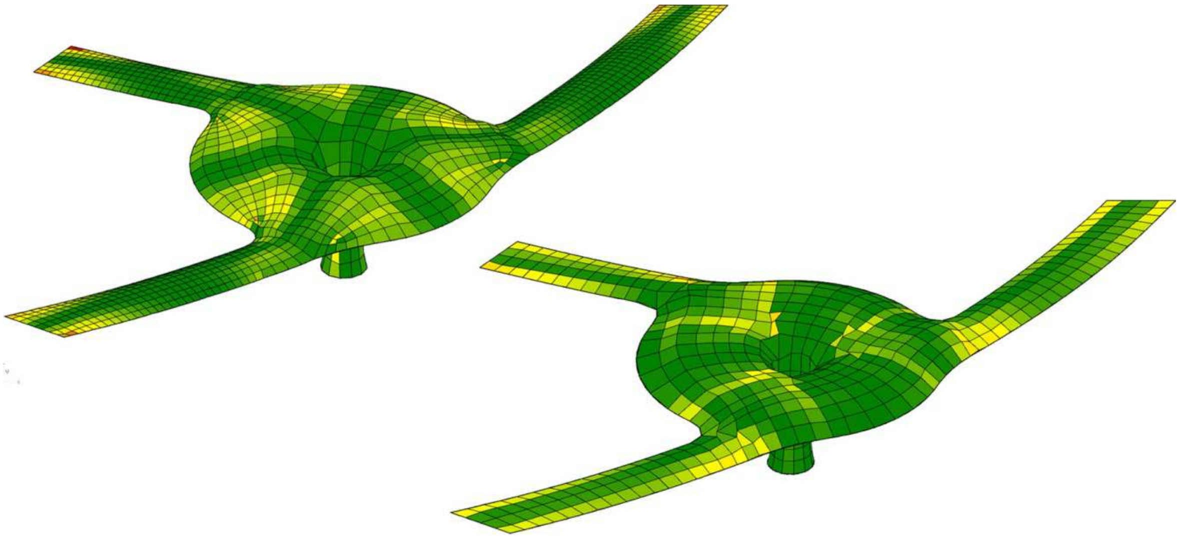

Figure 34 Comparison of deviation from planarity for different panel layouts, post-optimisation. Left: Mesh from principal stress field. The optimisation clearly caused the geometry to crease. Right: The mesh from the manually created base complex behaved even worse with multiple triangles forming at transition points of the geometry.

12 Conclusion

12.1 Summary

Methods for the generation of topological complexes for both triangle and quadrangular meshes, respecting the rigorous constraints in architectural roof and façade design, have been developed.

A unified framework has been implemented and integrated into the parametric design tool Grasshopper, making it a very accessible instrument within existing workflows. Based on the modular extension of a core mesh class, various methods of designing grid and panel layouts on discretised surfaces embedded in $\mathbb{R}^3$ have been effected.

In terms of programming, incremental remeshing tools have been written for generating isotropic base meshes, whilst parameterisation from manually produced base complexes permit control over number and placement of irregular vertices.

With a view of using vector fields as guidance for quadrangular remeshing, particular pre-processing steps have been incorporated in the software. These serve to simplify the topology of vector fields by reducing the number of singularities by smoothing and curl reduction, and to propagate them into ill-defined areas.

Streamline trace methods provide fast visual feedback on the quality of the above vector fields and can be used to inform quadrangular base complexes.

A periodic global parameterisation algorithm aligns wave functions to conjugate vector fields, resulting in the creation of quadrangular base complexes, from which anisotropic quad-meshes are extracted using a Morse-Smale complex.

Subdivision modelling tools have been enhanced for use with constraints specific to architectural design.

Finally, a modular dynamic relaxation suite was created permitting the definition of weighted multi-objective optimisation of meshes or curve networks. Information previously unavailable in Rhinoceros3d, such curvature data for meshes can be integrated as parameters in the problem statement.

In addition to the case study presented above, the various techniques reviewed and implemented have been applied to a number of actual building projects. The tools have been used for the generation of meshes inherited from the architect, form finding and constrained optimisation towards planarity of glass panels. The case-studies involved triangular and quadrangular meshes and were of such geometric complexity that the manual generation of fine meshes would not have been feasible.

12.2 Global and local methods

In many ways meshing in architecture inverts the priorities found in engineering or computer graphics applications. With view toward extended runtimes of projects, the immediacy of achieving a result is of less importance, and even the most exuberant building envelopes are usually rather simple when compared to the multitude of intrinsic features found in computer animation or 3d scanning. On the other hand, additional complexity arises from manufacture.

The aesthetic quality of the mesh, as well as the exact compliance with boundary conditions is paramount. Often, these boundary conditions are not global but local considerations which may be of very different nature – and at odds with the simple mathematical elegance of the global parameterisation techniques.
Though possible in theory, the formulation of numerous local constraints is far from instinctive with the global problem formulations presented in this thesis and other papers reviewed. Their reliance on numerous parameter values whose direct impact is difficult to judge make it less intuitive than local methods which often provide direct and instant visual feedback of changes to parameters. Local adaptability is therefore poor or must be performed once an initial mesh is found.

There are doubts about the exact reproducibility of a given result. With global methods, seemingly minor adjustments to global input parameters or geometry can have major repercussions on the suitability of vector and size field, convergence of the wave function, and as a result on location and number of irregular vertices in the final mesh. Such reproduction might become necessary only with slight changes to the input geometry, as are frequent in early design stages. In case of great changes to the output of a global optimisation method, down-the-line associations in a parametric workflow might corrupt, requiring their time-consuming re-establishing.

Some of these downfalls will undoubtedly vanish as methods improve, but there remains with the designer a distinct sensation of losing control over crucial design aspects. On the other hand global methods may indeed become a necessity for envelopes of great geometric complexity and ever tightening project schedules.

Architectural production differs from other industries in that the end-user product, i.e. buildings, is generally not mass-produced. Innovation occurs with almost every project, be it due to site specificity, client or architects ambition. At the same time, building components, or rather the manufacturing processes behind them are generally readymade – and building codes must be complied with.
This seeming contradiction is reflected in the digital design community. There exists powerful building-information-management software which at the time of designing also provides data for the institutionalised aspects of design, such as area schedules, cost estimates or tenders. Its power often falls short when extravagant architecture is attempted. To predict all requirements a designer might have is simply impossible. The digital design community is reacting by continuously providing small add-ins to existing software frameworks. Programs are decidedly bespoke, small and modular. Since only small modules must be programmed, such plug-ins, often truly project-specific, permit innovation without exceeding timeframes. Their uses in only single applications make rigorous debugging less of a requirement.
Itself making use of Rhinocommon, the mesh class written as part of this thesis allows many additional components to interface. In particular the dynamic relaxation suite was constantly expanded depending on project requirements. New components seldom required more than a handful lines of additional code but would then fit into the workflow seamlessly.
It should be noted therefore that the work presented in this thesis is far from conclusive. Much rather it is purposely left open-ended and ready for expansion as seen fit.

To answer this question as to the superiority of either global or modular approach in digital design is well beyond the scope of this thesis but perhaps it is right for now to claim once more an architectural “both-and”.

12.3 Further work

The paramount issue to be addressed is an improvement in the reliability of the wave alignment algorithm. The global periodic parameterisation did not always converge to a satisfactory outcome. As the success of the whole approach is dependent on successful convergence of the optimisation, this severely impeded its use when it came to working to tight project timelines. Supplying a suitable size-field seemed to correct this issue, additional vector and size field pre-processing steps may therefore be required to ensure more coherent results.

Furthermore, the work implemented in this thesis relates mainly to geometric processing. By virtue of an arbitrary vector field to guide the remeshing process the quad-meshing method presented here can be extended into the engineering domain, e.g. by using fields based on structural or otherwise informed concerns.

Finally, multi-resolution subdivision modelling may lend itself particularly well to structural optimisation procedures as the optimisation can be performed on a coarse base complex, exponentially reducing the number of variables whilst potentially retaining a linear relationship between the complexes.

13 References

ADRIAENSSENS, S., NEY, L., BODARWE, E. & WILLIAMS, C. J. K. Dutch Maritime Museum: Form-finding of an irregular faceted skeletal shell. International Association for Shell and Spatial Structures (IASS) Symposium 2009: Evolution and Trends in Design, Analysis and Construction of Shell and Spatial Structures, 2009 2009. 324-325.

ALLIEZ, P., COHEN-STEINER, D., DEVILLERS, O., LEVY, B. & DESBRUN, M. 2003. Anisotropic polygonal remeshing. *ACM SIGGRAPH 2003 Papers.* San Diego, California: ACM.

ANON 2008. Nord Park Cable Railway by Zaha Hadid Architects. *The Architect's Journal*.

AUTODESK 2012a. 3d Studio Max. 2012 ed.

AUTODESK 2012b. Maya. 2012 ed.

BARNES, M., TOPPING, B. & WAKEFIELD, D. 1977. Aspects of form finding by dynamic relaxation. *In:* UNIVERSITY, T. C. (ed.) *International conference on slender structures.* London: The City University.

BO, P., POTTMANN, H., KILIAN, M., WANG, W. & WALLNER, J. 2011. Circular arc structures. *ACM Trans. Graph.,* 30**,** 1-12.

BOMMES, D., LÉVY, B., PIETRONI, N., PUPPO, E., SILVA, C. T., M & ZORIN, D. State of the Art in Quad Meshing. Eurographics STARS, 2012 Cagliari, Italy.

BOTSCH, M., KOBBELT, L., PAULY, M., ALLIEZ, P. & LÉVY, B. 2010. *Polygon Mesh Processing*, A K Peters.

BRELL-COKCAN, S. 2007. *Supporting Structure for Freeform Surfaces in Buildings*. 12/308,617.

CARPO, M. 2011. *The Alphabet and the Algorithm*, MIT Press.

COLLETTI, M. 2010. Exuberance and Digital Virtuosity. *Architectural Design,* 80**,** 8-15.

SAWAPAN DESIGN, S. 2012. Millepede.

DIJKSTRA, E. W. 1959. {A Note on Two Problems in Connection with Graphs}. *Numerical Mathematics,* 1**,** 269--271.

DONG, S., BREMER, P.-T., GARLAND, M., PASCUCCI, V. & HART, J. C. 2006. Spectral surface quadrangulation. *ACM Trans. Graph.,* 25**,** 1057-1066.

EDELSBRUNNER, H., HARER, J. & ZOMORODIAN, A. 2001. Hierarchical morse complexes for piecewise linear 2-manifolds. *Proceedings of the seventeenth annual symposium on Computational geometry.* Medford, Massachusetts, United States: ACM.

EIGENSATZ, M., DEUSS, M., SCHIFTNER, A., KILIAN, M., MITRA, N. J., POTTMANN, H. & PAULY, M. 2010a. Case Studies in Cost-Optimized Paneling of Architectural Freeform Surfaces

Advances in Architectural Geometry 2010. *In:* CECCATO, C., HESSELGREN, L., PAULY, M., POTTMANN, H. & WALLNER, J. (eds.). Springer Vienna.

EIGENSATZ, M., KILIAN, M., SCHIFTNER, A., MITRA, N. J., POTTMANN, H. & PAULY, M. 2010b. Paneling architectural freeform surfaces. *ACM Trans. Graph.,* 29**,** 1-10.

FLÖRY, S. & POTTMANN, H. Ruled Surfaces for Rationalization and Design in Architecture. ACADIA 10: LIFE in:formation, On Responsive Information and Variations in Architecture, 2010 New York. ACADIA, 103-109.

FOUNDARTION, B. 2012. Blender.

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

HERTZMANN, A. & ZORIN, D. 2000. Illustrating smooth surfaces. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques.* ACM Press/Addison-Wesley Publishing Co.

HORMANN, K., POLTHIER, K. & SHEFFER, A. 2008. Mesh parameterization: theory and practice. *ACM SIGGRAPH ASIA 2008 courses.* Singapore: ACM.

HUANG, J., ZHANG, M., MA, J., LIU, X., KOBBELT, L. & BAO, H. 2008. Spectral quadrangulation with orientation and alignment control. *ACM Trans. Graph.,* 27**,** 1-9.

HWANG, K. & KNIPPERS, J. 2010. Stability of single layered grid shells with various connectors. *Structures & Architecture.* CRC Press.

INRIA 2010. Graphite.

KARNI, Z. & GOTSMAN, C. 2000. Spectral compression of mesh geometry. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques.* ACM Press/Addison-Wesley Publishing Co.

KHODAKOVSKY, A., LITKE, N., SCHR\, P., \#246 & DER 2003. Globally smooth parameterizations with low distortion. *ACM Trans. Graph.,* 22**,** 350-357.

KNIPPERS, J. & HELBIG, T. The Frankfurt Zeil Grid Shell. *In:* VALENCIA, E. D. L. U. P. D., ed. Symposium of the International Association for Shell and Spatial Structures, 2010 Valencia.

KOVACS, D., MYLES, A. & ZORIN, D. 2010. Anisotropic quadrangulation. *Proceedings of the 14th ACM Symposium on Solid and Physical Modeling.* Haifa, Israel: ACM.

KÄLBERER, F., NIESER, M. & POLTHIER, K. 2007. QuadCover - Surface Parameterization using Branched Coverings. *Computer Graphics Forum,* 26**,** 375-384.

LANQUETIN, S. & NEVEU, M. 2009. Reverse Catmull-Clark Subdivision.

LEVIN, A. 1999. Combined subdivision schemes for the design of surfaces satisfying boundary conditions. *Comput. Aided Geom. Des.,* 16**,** 345-354.

LEVY, B., PETITJEAN, S., RAY, N. & MAILLOT, J. 2002. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.,* 21**,** 362-371.

LEVY, B. & ZHANG, H. 2010. Spectral mesh processing. *ACM SIGGRAPH 2010 Courses.* Los Angeles, California: ACM.

LIU, Y., POTTMANN, H., WALLNER, J., YANG, Y.-L. & WANG, W. 2006. Geometric modeling with conical meshes and developable surfaces. *ACM SIGGRAPH 2006 Papers.* Boston, Massachusetts: ACM.

LIU, Y., XU, W., WANG, J., ZHU, L., GUO, B., CHEN, F. & WANG, G. 2011. General planar quadrilateral mesh design using conjugate direction field. *ACM Trans. Graph.,* 30**,** 1-10.

MARINOV, M. & KOBBELT, L. 2004. Direct Anisotropic Quad-Dominant Remeshing. *Proceedings of the Computer Graphics and Applications, 12th Pacific Conference.* IEEE Computer Society.

MATHWORKS 2012. Matlab. 2012a ed. Natick, Massachusetts.

MCNEEL 2011. Rhinoceros 3d 5.0 Beta. Seattle.

MCNEEL 2012. Grasshopper3d. 0.8 ed.

MEYER, M., DESBRUN, M., SCHRÖDER, P. & BARR ALAN 2002. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds.

NIESER, M., REITEBUCH, U. & POLTHIER, K. 2011. CubeCover– Parameterization of 3D Volumes. *Computer Graphics Forum,* 30**,** 1397-1406.

PALACIOS, J. & ZHANG, E. 2007. Rotational symmetry field design on surfaces. *ACM Trans. Graph.,* 26**,** 55.

PETER, S., ORGANIZERS PETER SCHR, O., PETER SCHR, O., DENIS, Z., TONY, D., DAVID, R. F., VB, W., LEIF, K. & MICHAEL, L. 2009. Subdivision for Modeling and Animation.

POLTHIER, K. & PREUß, E. 2003. Identifying Vector Fields Singularities Using a Discrete Hodge Decomposition. *Visualization and Mathematics***,** 21.

POTTMANN, H. 2008. Geometry of architectural freeform structures. *Proceedings of the 2008 ACM symposium on Solid and physical modeling.* Stony Brook, New York: ACM.

POTTMANN, H., LIU, Y., WALLNER, J., BOBENKO, A. & WANG, W. 2007. Geometry of multi-layer freeform structures for architecture. *ACM Trans. Graph.,* 26**,** 65.

POTTMANN, H., SCHIFTNER, A., BO, P., SCHMIEDHOFER, H., WANG, W., BALDASSINI, N. & WALLNER, J. 2008. Freeform surfaces from single curved panels. *ACM SIGGRAPH 2008 papers.* Los Angeles, California: ACM.

RAY, N., LI, W. C., L\, B., \#233, VY, SHEFFER, A. & ALLIEZ, P. 2006. Periodic global parameterization. *ACM Trans. Graph.,* 25**,** 1460-1485.

RAY, N., VALLET, B., LI, W. C., L\, B., \#233 & VY 2008. N-symmetry direction field design. *ACM Trans. Graph.,* 27**,** 1-13.

SAKAROVITCH, J. 2009. Gaspard Monge founder of "constructive geometry". *Third International Congress on Construction History.* Cottbus.

SCHIFTNER, A. & BALZER, J. 2010. Statics-Sensitive Layout of Planar Quadrilateral Meshes

Advances in Architectural Geometry 2010. *In:* CECCATO, C., HESSELGREN, L., PAULY, M., POTTMANN, H. & WALLNER, J. (eds.). Springer Vienna.

SCHLAICH, J. & SCHOBER, H. Design Principles of Glass Roofs. Lightweight Structures in Civil Engineering, 2002 Warsaw, Poland. International Association for Shell and Spatial Structures, 815-827.

SCHOBER, H. 1998. Netzkuppeln mit ebenen Maschen. Beispiel: Flußpferdehaus im Berliner Zoo. *Beratende Ingenieure,* 28**,** 5.

SCHOBER, H., Kürschner, Kai., Jungjohann, H. 2004. Neue Messe Mailand - Netzstruktur und Tragverhalten einer Freiformfläche. Stahlbau 73, Heft 8. Ernst & Sohn, Berlin

SCHRÖDER, P., ZORIN, D., DEROSE, T., FORSEY, D., KOBBELT, L. & LOUNSBERY, M. 2009. Subdivision for Modeling and Animation.

SHARP, B. 2000. *Subdivision Surface Theory* [Online]. http://www.gamasutra.com/view/feature/131585/subdivision_surface_theory.php: Gamasutra. [Accessed February 2012].

STAM, J. 1998. Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values. *Proceedings of the 25th annual conference on Computer graphics and interactive techniques.* ACM.

TAUBIN, G. 1995. A signal processing approach to fair surface design. *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques.* ACM.

TAUBIN, G. Geometric signal processing on polygonal meshes. Eurographics 2000, 2000 Interlaken, Switerland.

TRICOCHE, X. 2002. PhD, Universität Kaiserslautern.

VALLET, B. & LÉVY, B. 2008. Spectral Geometry Processing with Manifold Harmonics. *Computer Graphics Forum,* 27**,** 251-260.

WALLNER, A. & SCHIFTNER, M. 2010. Tiling freeform shapes with straight panels: Algorithmic methods. *In:* AL., C. E. (ed.) *Advances in Architectural Geometry 2010.* Hamburg: Springer.

WALLNER, J. & POTTMANN, H. 2011. Geometric Computing for Freeform Architecture. *Journal of Mathematics in Industry,* 1**,** 1-19.

WILLIAMS, C. J. K. 2001. The analytic and numerical definition of the geometry of the British Museum Great Court Roof. *In:* BURRY, M., DATTA, S., DAWSON, A. & ROLLO, A. J. (eds.) *Mathematics & design 2001.* Geelong, Victoria, Australia: Deakin University.

YACINE, B., UNIVERSITE DE, B., SANDRINE, L. & MARC, N. 2010. A Reverse Scheme For Quadrilateral Meshes.

ZADRAVEC, M. & SCHIFTNER, A. 2010. Designing Quad-dominant Meshes with Planar Faces. *Computer Graphics Forum,* 29**,** 8.

ZHANG, M., HUANG, J., LIU, X. & BAO, H. 2010. A wave-based anisotropic quadrangulation method. *ACM Trans. Graph.,* 29**,** 1-8.

ZORIN, D. 2006. Modeling with multiresolution subdivision surfaces. *ACM SIGGRAPH 2006 Courses.* Boston, Massachusetts: ACM.

14 Appendix A

A.1 Curvature estimates on triangle meshes

In the discrete setting, differential operators are less intuitive for it means reconstructing the properties of a smooth surface from a finite set of samples. Numerous methods have however been proposed to approximate intrinsic metrics from the piecewise linear mesh data itself. For an intuitive overview see (Botsch et al., 2010).

A.1.1 Mean and Gaussian Curvature Estimation on a triangle mesh

Differential properties are approximated over the local neighbourhood N_i of each mesh vertex $v_i \epsilon\ V$. The implemented software tool permits the selection of an arbitrary number of vertex loops around v_i, but areas larger than two loops significantly decrease performance without adding great improvements in terms of accuracy. Most commonly, the 1-loop around a vertex is sufficient.

The divergence of the gradient given by the discrete Laplacian (see chapter 4.1.5) is then integrated over this local averaging domain A_i such that

$$\Delta f(v_i) = \frac{1}{2A_i} \sum_{v_j \epsilon\, N_i} D_{ij}(f_j - f_i)$$

where

$$D_{ij} = (cot\alpha_{ij} + cot\beta_{ij})$$

and the angles α and β are the angles opposing edge e_{ij}. A_i is the Voronoi area around vertex v_i, defined as

$$A_i = \frac{1}{8} \sum_{j\, \epsilon\, N_i} D_{ij} \left\|p_i - p_j\right\|^2$$

This area measurement A_i, augmented to deal with obtuse triangles, is proposed by Meyer et al. (2002) as it leads to a coherent, non-overlapping tiling of the mesh surface.

Taking the vertex coordinates p_i as the input function $f(v_i)$, and taking half the magnitude of the resulting vector, yields the most commonly used approximation of mean curvature K_{H_i}used in mesh processing.

On the boundary of a surface, the mean curvature is ill defined. To obtain usable values at the surface boundaries, we proceed as follows: For each boundary vertex v_i, we sum up the vectors V_{ji} for interior neighbours v_j. The resulting vector V_g is then normalised by the number of contributing neighbours. A “ghost” vertex at the end of this vector is then added to the 1-loop around v_i, simulating its continuity.

A.1.2 Vertex Normal vector

The unitised mean curvature vector $\hat{v}_{MC_i}$ is taken to be the normal vector at the vertex. In planar regions, or where v_{MC_i} vanishes due to the cancellation of its contributing terms, the normal vector is assumed to be the normalised average of the adjacent face normal vectors v_{N_F}.

A.1.3 Gaussian curvature

To estimate the Gaussian curvature, the angles of the edges adjacent to a triangle are simply summed and subtracted from the value of a planar configuration, 2π

$$K_G(v_i) = (2\pi - \sum_{j \epsilon F_i} \theta_j)\,/A_i$$

Where $\theta_j$is the vertex angle in radians of the edges of each face f_j touching vertexv_i.
Again, a ghost vertex is introduced to notionally close the one-loop of boundary vertices.

A.1.4 Principal curvature directions

Given the values for mean and Gaussian curvatures, the magnitudes of the principal curvatures can already be found explicitly

$K_{1_i} = K_{H_i} + \sqrt{\Delta(p_i)}$ and $K_{2_i} = K_{H_i} - \sqrt{\Delta(p_i)}$ with $\Delta(p_i) = K_{H_i}^2 - K_{G_i}$

Principal curvature directions can also derived from the data (Meyer et al., 2002). The osculating circle and therefore the normal curvatures can geometrically be deduced in the plane defined by each edge e_{ij} and the vertex normal v_{N_i}. In the case of vertex valence greater than three, this leads to an over-determined system which can be minimised in the least squares sense, yielding the curvature tensor in the tangent plane.
The principal curvature directions are the eigenvectors of the above tensor, and can subsequently scaled by their respective magnitudes computed above (Meyer et al., 2002).