Mastering the "Sequential Roof"

Computational Methods for Integrating Design, Structural Analysis, and Robotic Fabrication

Aleksandra Anna Apolinarska, Ralph Bärtschi, Reto Furrer, Fabio Gramazio, and Matthias Kohler

A. A. Apolinarska, F. Gramazio, M. Kohler
Gramazio Kohler Research, ETH Zurich, Switzerland
apolinarska@arch.ethz.ch ✉
gramazio@arch.ethz.ch
kohler@arch.ethz.ch

R. Baertschi
ROB Technologies AG, Switzerland
baertschi@rob-technologies.com

R. Furrer
Dr. Lüchinger+Meyer Bauingenieure AG, Switzerland
rfu@luechingermeyer.ch

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_17, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

Abstract

This paper gives insight into a cross-disciplinary computational workflow developed and implemented in the recently completed "Sequential Roof" project at ETH Zurich. The project is a 2308 m^2 freeform, load-bearing timber structure consisting of nearly 50,000 members, robotically assembled layer-by-layer into trusses. The design and analysis of the highly differentiated structure required bespoke computational methods combined into an iterative process to solve the complex interrelations between geometry, structural behaviour, and fabrication constraints. Here, we describe this process, starting with (1) the geometric definition of the roof and (2) its structural model representation and evaluation with respect to the used connection method. Further we elaborate on (3) a randomised vertex population algorithm for the nail connection, and (4) the greedy algorithm to determine the necessary modifications. Ultimately, we explain how this computational workflow was implemented in the construction design phase of the project and discuss transferability of the approach and the architectural outcome.

Keywords:
computational design, computational geometry, robotic fabrication,
robotic assembly, timber construction, structural design, timber structures

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_17, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

1. Introduction

1.1 Project Description and Context

The "Sequential Roof" was developed by the group Gramazio Kohler Research at ETH Zurich for the new "Arch_Tec_Lab" building of the Institute of Technology in Architecture (ITA) at the ETH Hoenggerberg Campus. The building itself was planned and realised as a multidisciplinary research project (ITA 2016). The concept behind the roof's design was to investigate possibilities and constraints of computational design and robotic assembly on a full architectural scale (Willmann et al. 2015). Following previous, smallerscale experiments of the group, the design concept focussed on using ordinary, low-engineered softwood elements with a simple, linear geometry and notch-free joints, to create complex, versatile, and highly articulated structures (Gramazio, Kohler & Willmann 2014).

The roof structure, covering an area of 2,308 m^2, consists of 168 individual trusses, spanning maximum 14.70 m between steel box beams of the primary structure (Fig. 1) (Adam 2014). Each timber truss is composed of average of 370 geometrically unique timber slats, stacked in a layerwise alternating way and joined together by nails connecting each two overlapping slat ends. Using a robotic setup, the elements are fabricated and assembled sequentially in a fully automated process, where each slat is cut to size and then directly placed and joined with the rest of the truss structure (Fig. 19) (Apolinarska et al. 2015). Using simple elements that require minimal and fast processing (simple cuts) and focussing on full automation in assembly are key features that distinguish the project from other recently realised nonstandard timber structures, which make intense use of multiaxis CNC woodworking techniques to produce complex, curved elements with intricate connections and which are then assembled manually.

1.2 Computational Workflow

Such fabricationdriven design addressing a novel construction method radically challenges the conventional, phase-based process where the level of detail increases with each planning stage. Here, it was mandatory that design, analysis, and execution planning are tightly integrated and developed concurrently because of the obscure and complex dependencies between geometry, structural behaviour, and fabrication details. In consequence, it was not possible to generate a valid (structurally sound, feasible to fabricate, and architecturally correct) solution explicitly from a given set of input parameters.

With a "heavy" digital model, large data sets, long calculation times, and tight schedule to produce a final model ready to fabricate, our strategy was to start with a simple and mostly underdimensioned model, and to iterate through the loop of analyses and local modifications until all problems were resolved. The key challenge was therefore to establish an integrated workflow to facilitate this

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_17, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

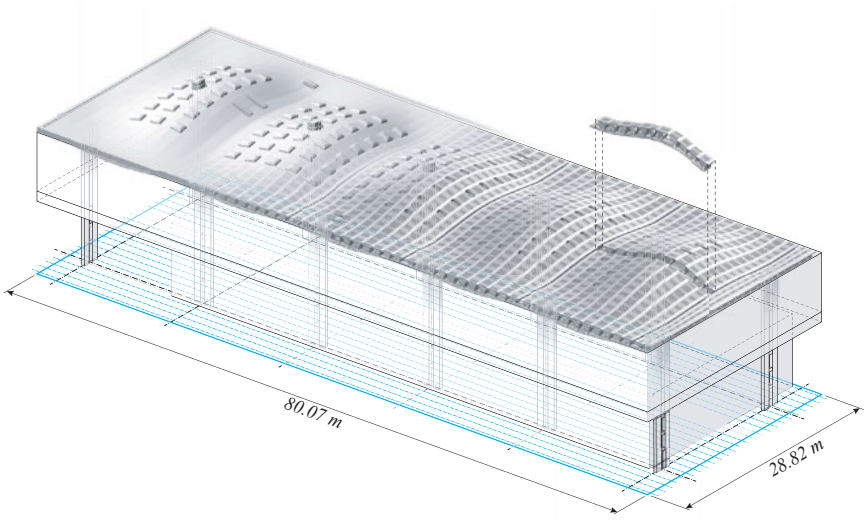

Figure 1. Isometric overview of the roof. It is composed of 168 timber trusses supported by a primary steel structure.

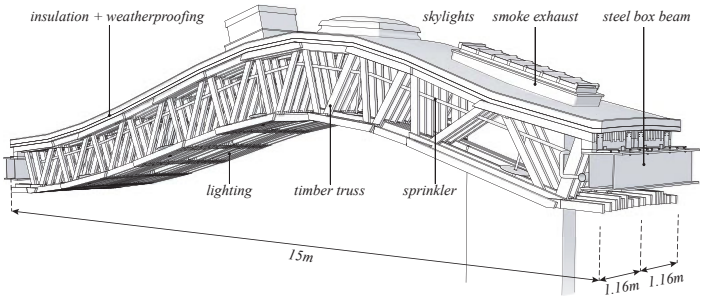

Figure 2. A pair of timber trusses with subsystems (sprinkler system, smoke exhaust, electrical, lighting, skylights). Insulation and weatherproofing layers are applied directly onto the structure, without additional boarding. Each truss rests on steel box beams, with one fixed bearing and the other movable in the longitudinal direction.

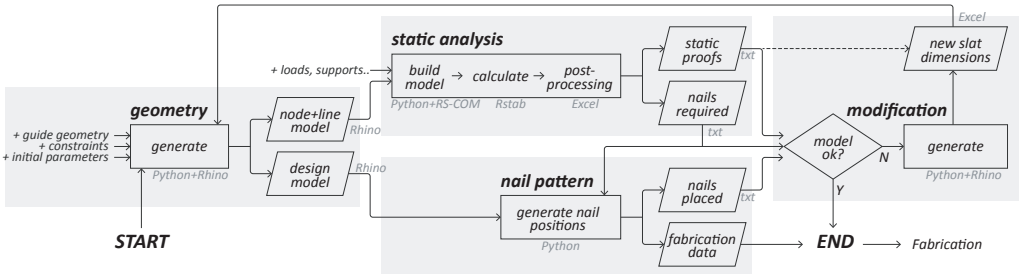

Figure 3. Schematic of the established computational workflow.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_17, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

process. In short, each iteration consisted of four steps: First, generate the model given the current parameters (see Section 2). Then, perform structural calculations and evaluation (see Section 3). Next, generate nail pattern for each connection, observing structural and fabrication constraints (see Section 4). Finally, assess the results of both simulations and perform modifications (see Section 5). Apart from several bespoke algorithmic methods, the workflow involved data management and data exchange, error-proofing and evaluation methods.

2. Geometric Design

2.1 Geometry Setout

The overall undulating form of the roof is defined layerwise by pairs of cubic Bézier curves. The curves are discretised by *setout points* P (Fig. 4), which are then used as direct and fix reference for the truss geometry. Their position is derived from a possibly uniform triangulation of the truss and locally compromised by interfaces with building components such as skylights, exhaust shafts, sprinkler pipes, or primary structure (Fig. 2). With setout points referring to the outer boundary of the roof shape, the position of a slat's axis line L depends on its width w (Fig. 5). The given *setout points* and widths create a geometric stencil for elements in a layer. Due to the free-form overall shape of the roof, in each layer the *stencil* is different, and there is a slight shift between slats in neighbouring layers.

2.2 Layering

While the *stencil* is needed to determine geometry of the elements (axes, node points, end cuts of each slat) in each layer, the layering pattern defines which elements actually occur in the truss – it defines the composition of chord and diagonal slats through all 23 layer of a truss. For example, structural logic imposes continuity of the top and bottom chord, which could here be achieved by concatenating the individual elements into a symmetrically layered belts (Fig. 6) (Apolinarska et al. 2015). The resulting pattern consists of a repeated sequence of three layers of chord slats followed by a layer of diagonal slats. Local exceptions to this pattern occur when some slats need to be removed, for example, below the skylights (to let more light in) or to let smoke exhaust shafts through the roof structure (Fig. 2). In general, the geometric setup is relatively flexible, and its principles could easily be adapted to use in other projects.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_17, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

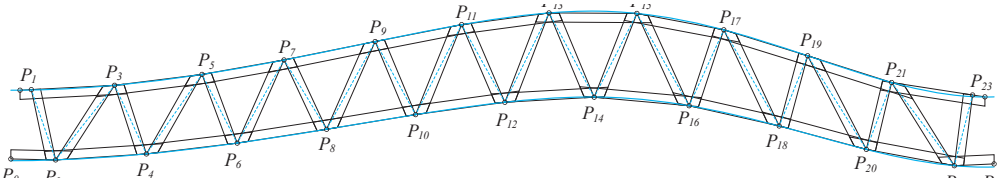

Figure 4. Setout geometry for a layer: a pair of Bézier curves, setout points P and stencil geometry.

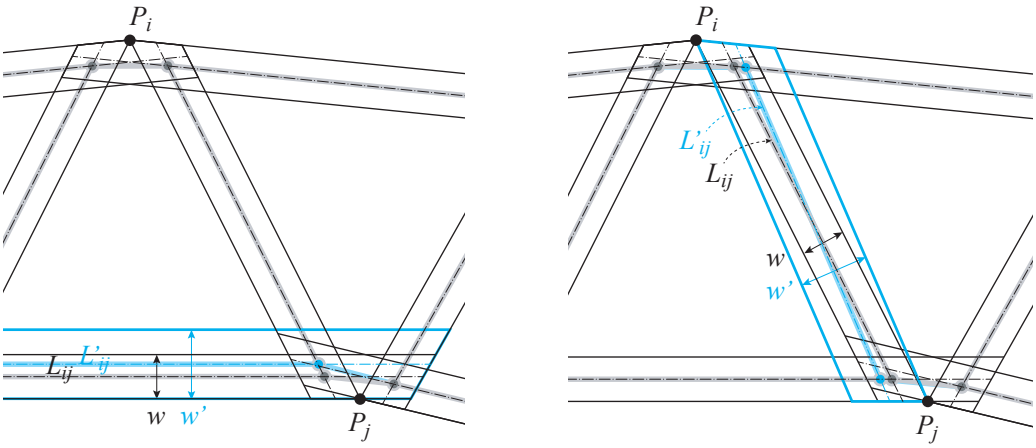

Figure 5. Detail of the geometric stencil. With setout points P fixed, the position of the slat's axis L depends on its width w.

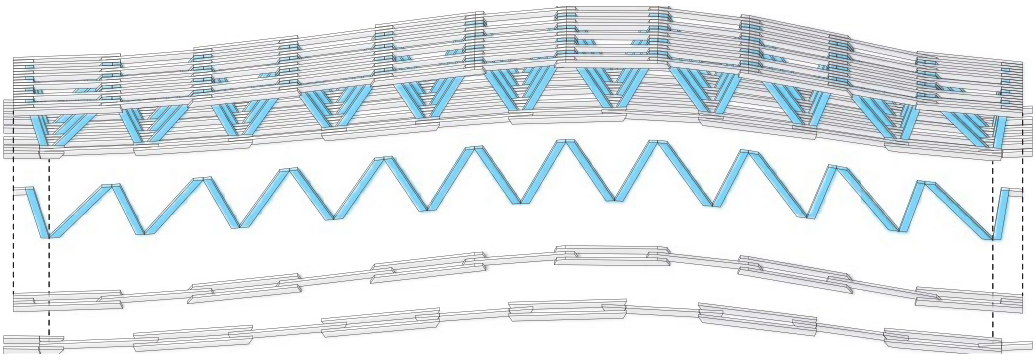

Figure 6. Layering pattern. (Apolinarska et al. 2015).

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016

3. Structural Analysis

3.1 Structural System

Through the layered buildup, the structural system of the timber trusses differs slightly from that of a typical truss, where the axes of diagonal and chord members usually intersect at nodal points. This is not the case here (Fig. 7), and the resulting eccentricity causes considerable shear forces and bending moments. Additionally, the load flow in cross direction (which is also wood's weak direction) is diverted, because the axes of connecting member do not intersect as they lie in different layers. Thus, the connections are modelled as *connector*-beams. Further, *connector*-beams have to be geometrically decomposed into segments so that the *connector* is normal to the shear plane (Fig. 8) to get the right section forces. Together, all these phenomena were challenging to represent and interpret with a beam statics program.

3.2 Analysis

The structural analysis process consisted of three steps: model setup, calculation, and post-processing. The model setup included acquiring the geometry (node points and members) into a structural analysis software and defining loads, load cases, supports, cross-sections, etc. From the calculation results, the internal forces were post-processed to produce twofold output information. The first output information was derived from beam proofs based on buckling, bending, normal and shear forces, and indicated which slat had to be wider to satisfy these. The second output information was the required number of nails per *connector* as a result of the internal forces in the *connector* beams.

3.3 Connection

The specific structural and fabrication logic of the structure required a connection method that would be both geometrically flexible and fully automatable. Further, because the static system of the coupled trusses is statically highly indeterminate, ductile connections were necessary to transfer loads – a brittle failure of connections could lead to a progressive collapse of a truss. Therefore, a connection technique using 90 mm long grooved nails, with a shaft diameter of $d = 3.4$ mm, proved to be both the simplest and the most applicable solution for this project. Automated nailing is a fast, cost-effective, and well-established CNC technology in timber construction. Compared to bolts, screws, or bulldog connections, resistance of one nail is quite smal; however, due to their relatively small diameter nails can be placed closer together and fit better in the connection's overlap area. In result, the sum of the nails can create enough resistance, more than the alternatives. The downside of this connection technique is the exceptional complexity that arises from combining multiple layers with different geometries.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016

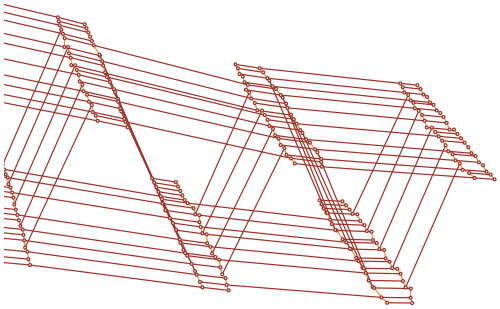

Figure 7. Point-and-line model generated for Rstab input.

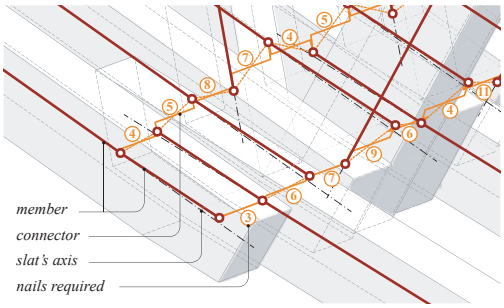

Figure 8. Detail of point-and-line model. The number of nails required is given individually for each connectorbeam. Through the eccentric connection, chord slats are represented by three (in general not colinear) members.

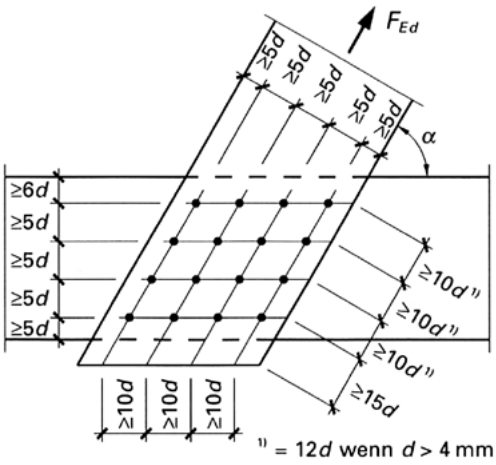

Figure 9. (left) Minimal distances for nailed connections according to the Swiss timber code (SIA 265:2003 Timber Structures 2003). For this project, both slats are considered loaded, so the edge offset are 6d (cross to fibre) and 15d (along fibre). Distances between nails are 5d and 10d respectively.

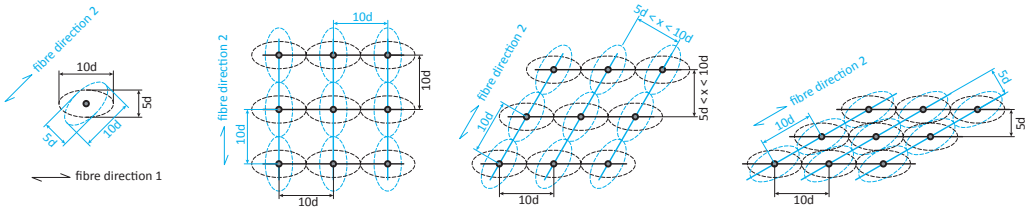

Figure 10. Interpretation of the spacing rules as elliptic lockout areas and comparison with a rhombic grid layout.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016

The Swiss timber code (SIA 265:2003 Timber Structures 2003) specifies rules for nailed connections, including minimum distances to slats' edges and between nails, depending on the fibre direction and the nail's shaft diameter d, as represented by a rhombic grid in Fig. 9. This distribution, however, is inefficient if multiple slats are overlapping with different angles between them. Instead, the distances were interpreted as a pair of elliptic lock-out areas around each nail, with long axes of the ellipses aligned with fibre directions of the two corresponding slats (Fig. 10). The edge offsets, resulting in a polygonal *feasibility region*, were calculated individually for each pair of connecting slats.

Still, the nails in each connection had to observe nails from the layer above and below. With a total of 129,840 *connector*-beams, each with individual geometric conditions, distribution of all the required nails in a reasonable, efficient, and compliant way could only be solved with computational methods. The next section describes the algorithm developed for this purpose.

3.4 Testing

The various assumptions mentioned above needed to be refined and confirmed by physical tests. First, small specimens consisting of three slats were tested to determine shear and bending stiffness parameters for connectorbeam elements in the calculation model. Also, connections with asymmetrically distributed nails were tested to determine the impact of symmetry on the performance of the connection – strongly asymmetric distribution of nails increases the risk of splitting of wood fibres due to onesided lateral tension forces. Finally, 15 full-scale trusses were load tested to get further assurance in regard to statics, and production processes. With the use of statistics the failure mode and the corresponding load could be predicted quite precisely.

4. Nail Placing Algorithm

4.1 Problem Description

Given the geometry of connecting slats, the challenge for the nail-placing algorithm is to find a solution to how to distribute the required number of nails in the connections in a way that is compliant with the spacing rules described in the previous section. Moreover, the nails should be distributed evenly (symmetrically). Also, a fabrication requirement is that each slat has at least one *fix-nail* at each end – a nail that lies exactly on the slat's axis line. As the trusses are assembled by a robot, the first nail has to be placed when the gripper holding the slat is still closed in order to fix the slat in place precisely. The fact that the gripper is still holding the slat in that moment constraints the *fixnail's* position (Fig. 20).

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_17, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

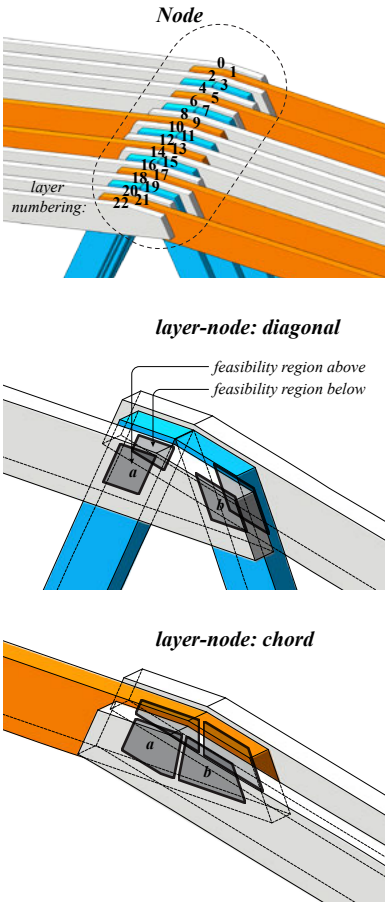

Figure 11. Node and layer-node.

```
Def VertexPacker(AlreadyPlacedNails, GeometricalSituation, RequiredNails):
     Generate the initial feasibility areas
    Loop as long as there is some nonempty feasibility area:
        Choose either upper or lower connector
         Pick m random vertices on the border of the feasibility area
        Discard vertices harming symmetry
        Pick best vertex
        Place a nail
        Recalculate both feasibility areas affected by this nail

counter = 0
solutions = []

While counter < 15:
    counter += 1

    If counter==3 and not solutions:
        DeleteRedundantNailsOnDifferentLayers(AlreadyPlacedNails)

    packing = VertexPacker(AlreadyPlacedNails, GeometricalSituation, RequiredNails)

    If enough_nails:
        result = [#redundant_nails_upper + #redundant_nail_lower, symmetry, packing]
          solutions.append(result)
```

Figure 12. Nail placing algorithms pseudocode.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016

4.2 Splitting the Problem into Sub-problems

Each *node*, understood as the sequence of connections of 23 layers of slats, refers to 22 pairs ("a" and "b") of *connectors*. The nail-placing problem is independent for each *node*. The problem of placing nails in the entire *node* can be split into a set of *layer-node* nail problems (Fig. 11). By *layer-node* we denote an abstraction that describes all nails going through a certain slat in the *node*, i.e. those placed in this slat (lower *connector*) and those from the slat above (upper *connector*).

There are two types of connections: between a chord slat and a diagonal, and between two chord slats. The nails inside diagonal slats (in diagonal *layer-nodes*) are placed first (layers 3-4, 7-8, 11-12, 15-16, and 19-20). The remaining nails are placed by solving for the chord slat in the middle of the belt (chord *layer-nodes* in layers 1-2, 5-6, 9-10, 13-14, 17-18, and 21-22) – this task is partially constrained by the already placed nails from the diagonal *layer-nodes*.

4.3 Random Vertex Population

Our first studies to solve the nail spacing problem were based on physical systems, for example, floating circles (Hockney & Eastwood 1981). However, it soon became apparent that the problem is too stacked to get a good floating behaviour. Also, the nail spacing task is not an optimisation problem, but one aiming at finding a feasible solution.

Therefore, an alternative approach was pursued: Nails are added to the solution one by one, placed on the border of the *feasibility region*. At the beginning, the *feasibility region* is constructed by edge offsets of the overlapping slats. When a nail is placed, and a corresponding nonconstant offset of its lockout zone is subtracted (Fig. 13). This means that the resulting new *feasibility region* may as well be disjoint. Experience shows that Boolean clipping in 3D CADsoftware lacks robustness (Schirra 2000). In order to get a robust polygon clipping behaviour, we decided to use a 2D polygon clipping library based on Vatti's algorithm (Vatti 1992; Murta 1997). For this, the lock-out zone ellipses are represented by a polygon with 72 vertices. The resolution of the ellipses, and of the resulting shape of the *feasibility region*, is an important parameter in the subsequent algorithm (Fig. 12).

The nail-placing procedure starts by finding a feasible fixnail configuration for all layers. After that, the remaining nails are distributed. For each layernode there are two feasibility regions, so first we choose whether the new nail should be placed in the upper or lower one. If both *feasibility regions* are non-empty, let $\Delta := n_R - n_P$ (where n_R = number of nails required, n_P = number of nails placed). If both connectors do not yet have enough nails (i.e. $\Delta_u > 0 \wedge \Delta_l > 0$), then the lower of the ratios A/Δ (where A = area of the feasibility region) wins; otherwise, the higher of ($\Delta_u >$, Δ_l) wins. Next, we randomly choose from vertices on the border of the selected *feasibility region*, restarting the algorithm a few times

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016

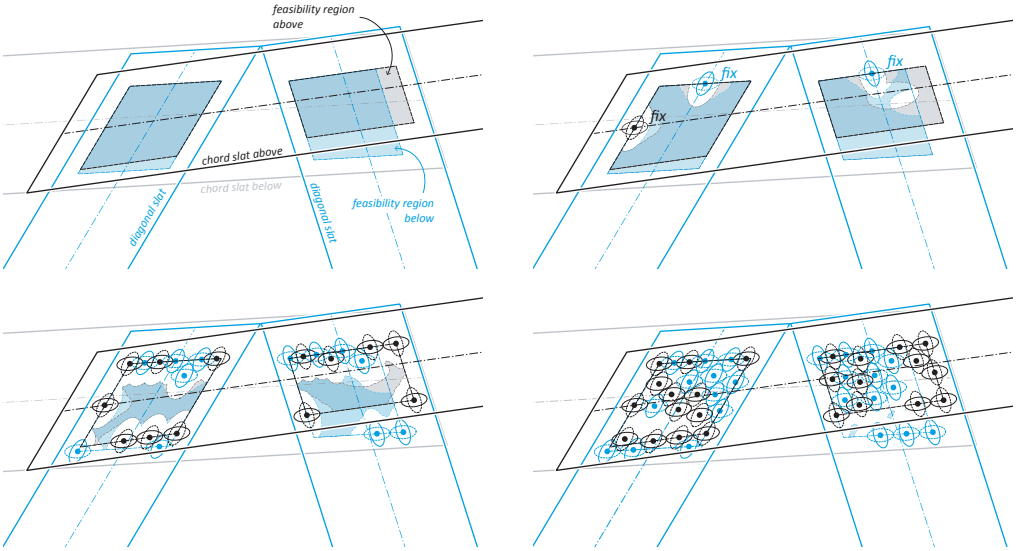

Figure 13. A step-by-step example of solving a diagonal layernode. Top left: Initial situation. Shaded areas denote the initial feasibility regions created by edge offsets. Top right: Fixnails placed at axis lines. New remaining feasibility regions after subtracting the offsets of lock-out areas, including those from nails from other layers not shown here. Bottom left: Situation after 10th iteration. Lockout areas of the same colour cannot intersect. In the lockout areas of different colours, only the ellipses in the layer they share cannot intersect, which is here the layer of the diagonal slats (ellipses aligned with axes of diagonals). Bottom right: Situation near completion. On the right there is still a non-empty feasibility region where two more nails could be placed.

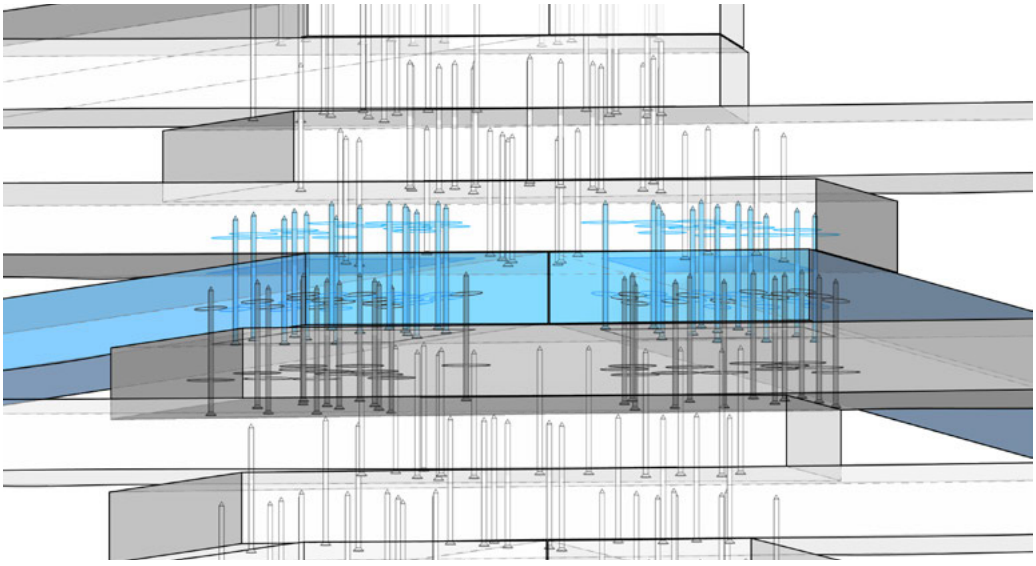

Figure 14. Final distribution of nails in a diagonal layer-node from the example in Figure 13 (isometric view from top).

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016

with different seeds. Using randomness helps to avoid bias and thus proved to improve the overall behaviour of the algorithm. Next, we evaluate the randomly selected candidates in terms of symmetry and discard solutions below the acceptable threshold. The symmetry of a nail pattern with respect to a slat is defined as the average of signed distances of the nails to the axis line. From the remaining candidate points, we pick the best solution to place a nail. Alternatively, one could restrict the choice of the random vertex to the convex hull of the *feasibility region*. This would automatically solve the dependence on the resolution.

The nail-placing algorithm iterates as long as the *feasibility regions* are non-empty. It may not be able to place all the required nails, or it may fail to place a fix-nail, or it may be able to place more nails than needed. This can cause problems: Redundant nails placed when solving for diagonal slats can impair the completion of the vertex population algorithms in the chord slats. To avoid this, if no solution was found after two trials, redundant diagonal nails which intersect the considered *feasibility region* are deleted. A surplus of more than 50% of required nails is also economically undesirable. The superfluous nails above the +50% threshold are therefore deleted. The selection is done carefully to maximise the symmetry in distribution around slat's axis.

5. Modification Strategies

5.1 Challenge

Both structural analysis and nail-placing procedure can yield negative results, meaning that the model needs some modification. Since the problems proved to be highly differentiated, no generalised solutions could be found. Also, as the overall form of the structure and the setout points were overconstrained, possible spectrum of modifications was confined to dimensioning (incrementing the slat's width) and geometric details (extending the slat's end cut) (Fig. 15). The choice of slat sizes was limited to three: 115 x 50 mm, 140 x 50 mm and 180 x 50 mm, which is a compromise between the necessary differentiation and economy of the fabrication process.

The problems of the first type – the failed beam proofs (as described in Section 3) – are explicit and non-negotiable. They are treated in the first place by changing the indicated slats' sizes as required by structural analysis. The second type of problems concerns the connection: Either some of the required nails or a fixnail could not be placed. In general, the solution is to change (usually increase) the feasibility area of the connection. These problems are far less trivial to solve. One of the reasons is that the problem does not explicitly indicate which element to modify – each connection is shared by two elements and in most cases changing only one of them is sufficient. Also, increasing the size of a slat to solve a problem at one end changes the situation in all of its connections at

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016

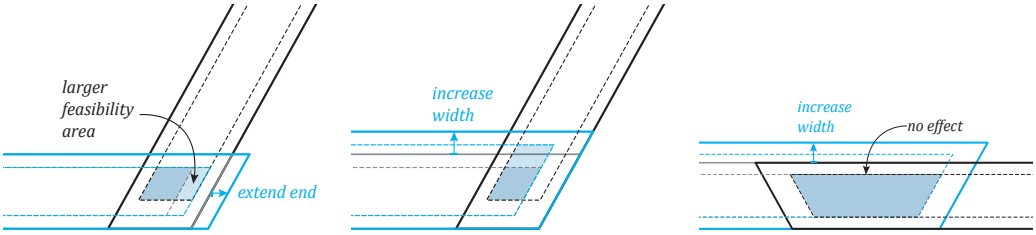

Figure 15. Two possibilities of modifying the overlap zone geometry: by extension and by increasing the size of a timber element. Extensions are preferred for the sake of material economy, but cannot be applied to diagonal members (as it would increase the eccentricity) and were not allowed in lower chord (for visual reasons) and certain further exceptions. Increasing size in chord-chord connections may be ineffective. (Apolinarska et al. 2015).

the other end too. Moreover, the effect of the modification cannot be evaluated exactly; this would require repeating the structural and fabrication analysis for each considered choice – technically possible, but inefficient. The actual number of nails that fit into the initial *feasibility region* cannot be directly deduced from its area (because it depends on the situation in other layers) – an estimate must suffice instead. In some cases, the modification does not increase the overlap area at all (Fig. 15). Again, modifying the size of a slat changes its axis and moves the structural *node* points.

5.2 Method

The solving procedure can be applied to each truss individually, which considerably reduces the complexity and speeds up the process. For each truss, first we try to solve all fixnail problems. Fortunately, they are rare, and their cause is easy to identify and eliminate. Next, we try to solve the "missing nail" problems, starting with the most economical method: extending the slat's end. For each *connector* there is only one possible slat to extend, and the problematic *connectors* are treated in the order according to the number of missing nails.

For each modification, the new *feasibility region* is calculated to check if its area really increased. For any applied modification, areas of all affected (shared by the slat) *feasibility regions* must be recalculated to estimate the number of nails that they could additionally accommodate and to add this amount to each connector's number of nails already placed ($n_P + n_A$).

The remaining "missing nail" problems are solved by enlarging the slat's width. Here, to establish a solving order, we chose a "greedy algorithm" mechanism in which the "worst" elements are treated first. The "worst" slat is determined with a heuristic approach by combining various parameters to sort all slats according to four criteria in the following sequence: difficulty (S_1), then the sum of all nails required (n_R) in all connectors (c) attached to a slat ($S_2 = -\sum_{i=1}^{c} n_{iR}$),

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016

followed by slat's current size (S_3=w), and finally nails overload (S_4), which is a ratio of the sum of missing nails (n_M) to the required nails (n_R): $S_4 = \sum_{i=1}^{c} n_{iM} / \sum_{i=1}^{c} n_{iR}$. Difficulty ($S_1$) is a sum of three values: sum of all nails missing in all its *connectors* ($\sum_{i=1}^{c} n_{iM}$), nails missing (n_{kM}) in slat's worst *connector* (*connector* with the highest number of nails required, and nails missing (n_{qM}) in slat's worst *node* (node with highest number of nails required in its *connectors*): $S_1 = \sum_{i=1}^{c} n_{iM} + n_{kM} + n_{qM}$. If the worst slat cannot be modified (because it has already maximal size) or the modification is not effective, try other slats connected to its worse end. After each modification, we sort the list of elements again and remove all "resolved" ones. The process iterates until the list is empty.

5.3 Results

Although each truss was processed individually (see example in Fig. 16) and problems were considered on a local level, the algorithm yielded consistent solutions throughout all trusses, producing similar patterns in similar trusses (Fig. 18), thus proving to be sufficiently robust to minor differences in input parameters, such as rounding errors. In terms of efficiency, the number of problems reduced by over 95% between iterations (Fig. 17). Most of the trusses were cleared after already three to four iterations, though, some required as much as seven iterations. With one full iteration costing almost 24 hours of work and calculation time, solving the model with as few iterations as possible was favourable. At the same time it is difficult to conjecture whether a method converging more slowly would yield a solution that is more efficient in terms of material consumption, because with the given slat sizes the size increment is often larger than needed, especially for minor problems. In the final model, the wood volume increased by +13% to 385 m³ compared with 339 m³ of the initial model. Nevertheless, if the roof had to be realised with one size only and dimensioned to the worst case, the increase would have been an estimated +59% (539 m³). It is worth pointing out that, as it is often the case in timber structures, also here the connections were the driving factor in dimensioning of the slats. In total, 815,984 nails were placed, with an overhead of +45.73% making use of the surplus area to increase the stiffness of the connections and improve the transferring loads.

6. Implementation

The entire roof model consisted of 49,858 slat elements (represented by extrusions, axis lines, outlines) generated based on 3,808 pairs of input curves. The structural representation involved 135,840 node points, 91,286 members, and 129,840 *connector*-beams. Handling such a "heavy" and differentiated model

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016

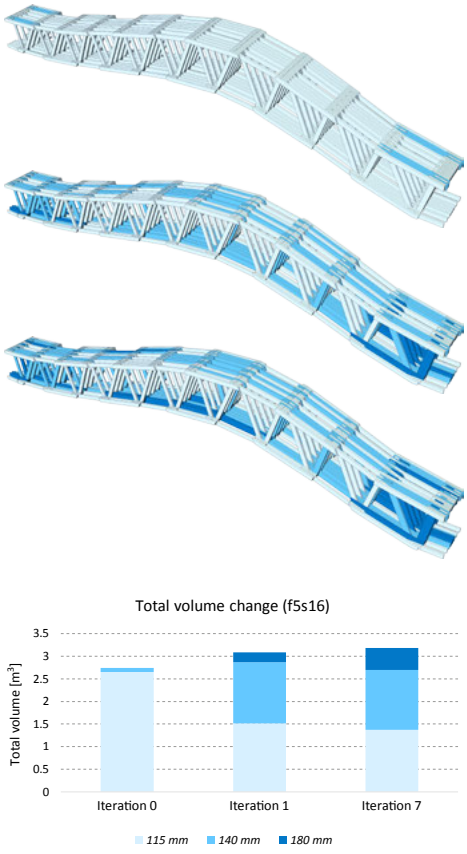

Figure 16. Solving steps for an exemplary truss: an initial model, after first and after last iteration (Apolinarska et al. 2015). Corresponding resulting wood volume with respect to the three slat sizes: 115–50 mm (light blue), 140–50 mm (blue), 180–50 mm (dark blue).

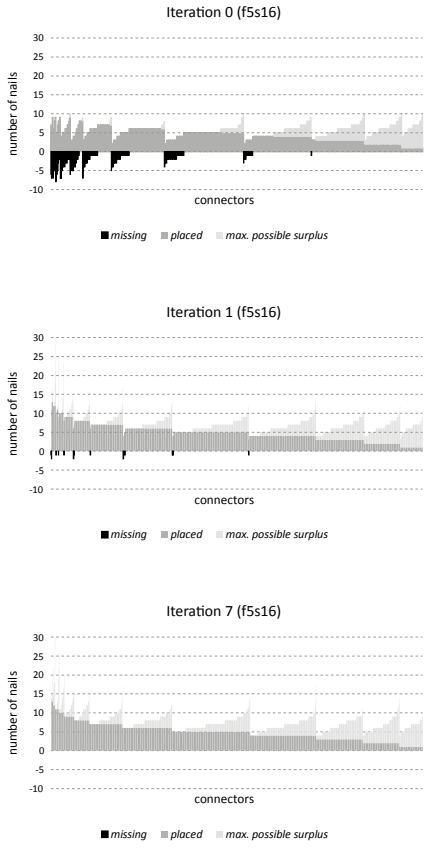

Figure 17. Solving steps for an exemplary truss. Number of nails missing (black), placed (grey) and maximal possible surplus (light grey) at different iterations.

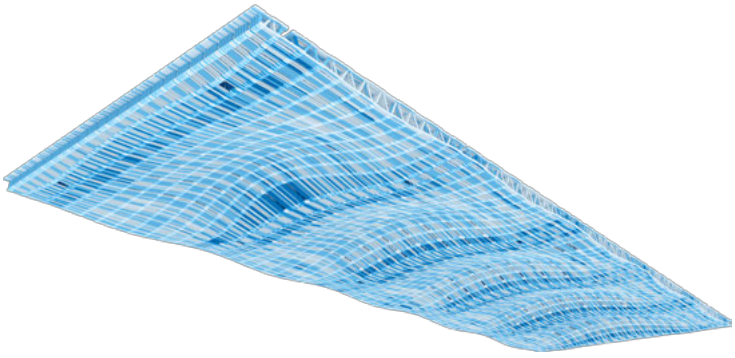

Figure 18. Final, resolved model of the entire roof structure, colour-coded according to the slat's width (see legend Fig. 16) (Apolinarska et al. 2015).

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016

required a high degree of automation in modelling and an efficient and intuitive data management strategy for query, survey, and control.

The architectural geometric model was generated using custom-made libraries based on RhinoCommon SDK of the 3D Modelling software McNeel Rhinoceros©. For the structural analysis in the software Dlubal Rstab©, the model was assembled with bespoke scripts using API modules that directly access the application (RS-COM) to overcome the hurdle of manually setting the extremely high amount of individual properties for each element. The post-processing of the calculation results was carried out in Excel with help of VBA macros.

The computational workflow relied heavily on intense exchange of large amounts of data, and the output format of the processed data at different steps of the workflow could be tailored accordingly, be it a 3D model or a data-set in text or spreadsheet format. Conventional representation methods such as 2D plans, elevations, and sections were mostly unfit to portray the relevant information.

Needless to say, error-proofing demanded special consideration, and cross-checking procedures had to be established. For example, the development of the highly complex nail pattern algorithm required an independent control script to examine the reported nail pattern results with the geometric solution, by redrawing the fibre-aligned ellipses and checking if they are collision-free. Additionally, at all stages the geometry and calculations were inspected at random, including visual control, and checked for consistency.

As mentioned earlier, the final, completely resolved model provided not only all calculations and detailing, but also output data for fabrication (Fig. 3). This feature is a radical difference to the conventional execution planning process and a major step forward to a complete, gapless digital chain. These output data were then converted into machine code of the large scale 6-axis gantry robot with which the roof trusses were built (Kramer 2016) (Fig. 19).

7. Conclusions

The project exemplifies the specific *modus operandi* needed for fabrication-driven design, which requires a concurrent collaboration between disciplines that are usually involved at different project stages (design, detailing, structural analysis, fabrication, and execution planning). It also highlights the importance of teamwork as soon as challenges and risks faced by planners and stakeholders go beyond the standard code of practice.

Overall, although many of the developed methods are very case specific, the core ideas of the project, i.e. the computational workflow, should be easily transferable to future projects. Still, the established computational framework holds a lot of potential for further development, for example by tighter integration of programming interfaces to simplify data exchange.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_17, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html

Figure 19. Automated fabrication and assembly using a 6-axis gantry robot (ERNE AG Holzbau) (Willmann et al. 2015).

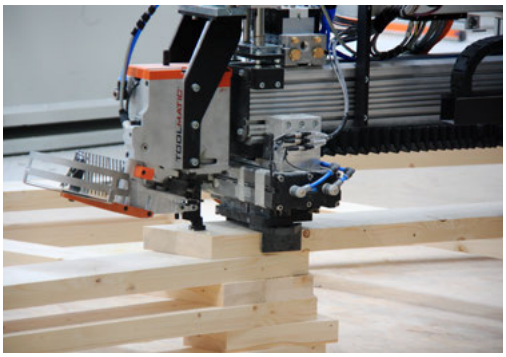

Figure 20. Fixing a slat with a fix-nail. After all slats in the layer have been placed and fixed, all remaining nails are shot.

Figure 21. Interior view of the completed roof.

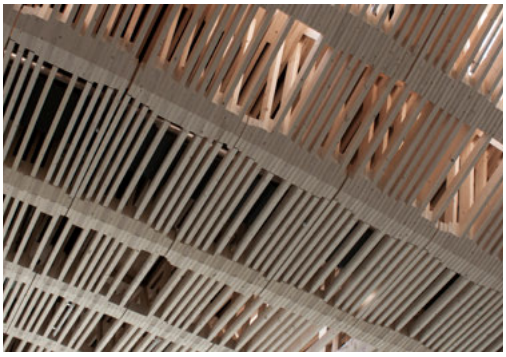

Figure 22. Detail soffit view of the completed roof. The varying width of the diagonal slats manifests itself in the connections. The shadow gaps between the timber trusses allow for building tolerances and shrinking and swelling of wood.

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016

Acknowledgements

The authors would like to thank the collaboration partners of the "Sequential Roof" project for both giving the opportunity of realising such an experiment and their generous support in the overall project. This includes the following realisation and consulting partners: overall planning: Arch-Tec-Lab AG (Guido Züger); structural engineering: Dr. Lüchinger+Meyer Bauingenieure AG; timber engineering: SJB Kempter Fitze AG; manufacturing and realisation: ERNE AG Holzbau; digital integration and fabrication control: ROB Technologies AG; structural design consultancy: Prof. Dr. Josef Schwartz (ETH Zurich); timber structure engineering consultancy: Prof. Dr. Andrea Frangi (ETH Zurich). Much of the "Sequential Roof" project would have not been possible without the valuable support of the Institute of Technology in Architecture (ITA) and ETH Zurich, which, in fact, initiated and supported this exciting endeavour. We also thank our Gramazio Kohler Research colleagues Michael Knauss and Jaime de Miguel, who contributed greatly to this project in its earlier phases.

References

Adam, Hubertus. 2014. "Holzkonstruktionen, Digital Fabriziert." *Zuschnitt – proHolz Austria*. Accessed at http://www.proholz.at/zuschnitt/53/holzkonstruktionen-digital-fabriziert/

Apolinarska, Aleksandra Anna, Michael Knauss, Fabio Gramazio, and Kohler Matthias. 2016. "Arch_Tec_Lab Roof." In *Advancing Wood Architecture*. London: Routledge. (in press)

Gramazio, Fabio, Matthias Kohler, and Jan Willmann. 2014. *The Robotic Touch – How Robots Change Architecture*. Zurich: Park Books.

Hockney, R. W., and J. W. Eastwood. 1981. *Computer Simulation Using Particles*. New York: McGraw-Hill.

ITA. 2016. "Arch_Tec_Lab." Accessed March 30 at http://ita.arch.ethz.ch/index.php/de/arch-tec-lab

Kramer, Martin. 2016. "Individual Serialism Through the Use of Robotics in the Production of Large-Scale Building Components." In *Robotic Fabrication in Architecture, Art and Design*, 460–67. Cham: Springer International Publishing.

Murta, Alan. 1997. "GPC – General Polygon Clipper Library." Accessed at http://www.cs.man.ac.uk/~toby/alan/software/

Schirra, Stefan. 2000. "Robustness and Precision Issues in Geometric Computation." In *Handbook of Computational Geometry*, edited by J.-R. Sack and J. Urrutia, 597–632. Amsterdam: Elsevier.

SIA 265 : 2003 Timber Structures. 2003. Switzerland. Accessed at http://www.webnorm.ch/normenwerk/ingenieur/sia265/e/D/Product

Vatti, Bala R. 1992. "A Generic Solution to Polygon Clipping." *Commun*. ACM 35 (7). New York, NY, USA: ACM: 56–63.

Willmann, Jan, Michael Knauss, Tobias Bonwetsch, Aleksandra Anna Apolinarska, Fabio Gramazio, and Matthias Kohler. 2015. "Robotic Timber Construction – Expanding Additive Fabrication to New Dimensions." *Elsevier: Automation in Construction*, 16–23. doi:10.1016/j.autcon.2015.09.011

S. Adriaenssens, F. Gramazio, M. Kohler, A. Menges, M. Pauly (eds.): Advances in Architectural Geometry 2016
 DOI 10.3218/3778-4_17, ISBN 978-3-7281-3778-4
http://vdf.ch/advances-in-architectural-geometry-2016.html