# Shape Structuralizer: Design, Fabrication, and User-driven Iterative Refinement of 3D Mesh Models

**Subramanian Chidambaram***
School of Mechanical Engineering
Purdue University
West Lafayette, IN, USA
schidamb@purdue.edu

**Yunbo Zhang***
Department of Industrial & System Engineering
Rochester Institute of Technology
Rochester, NY, USA
ywzeie@rit.edu

**Venkatraghavan Sundararajan**
School of Mechanical Engineering
Purdue University
West Lafayette, IN, USA
sundara5@purdue.edu

**Niklas Elmqvist**
College of Information Studies
University of Maryland
College Park, MD, USA
elm@umd.edu

**Karthik Ramani**
School of Mechanical Engineering
Purdue University
West Lafayette, IN, USA
ramani@purdue.edu

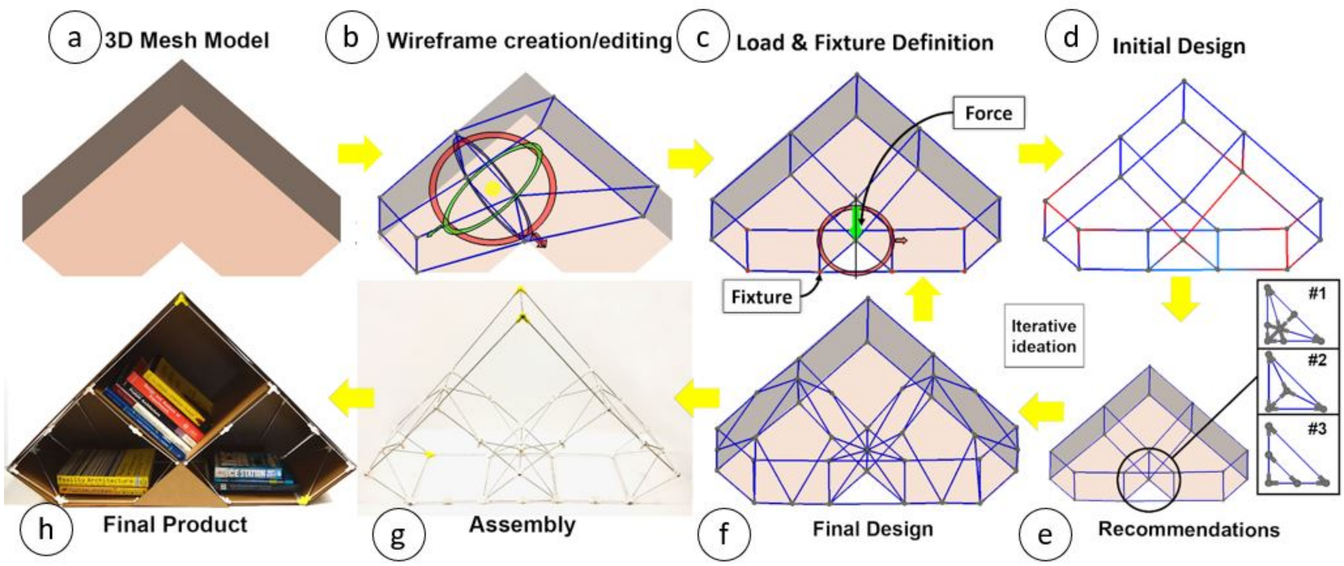


**Figure 1: Basic Shape Structuralizer workflow for a bookshelf design fabrication example. (a) The user inputs a 3D mesh model into Shape Structuralizer, and (b) quickly generates a rough shape using the wireframe editing tool. (c) After a preliminary scaffold is obtained, force load and boundary conditions can be added onto vertices via an interface. (c) Then the analysis module of Shape Structuralizer takes the design, load and fixture information, and (d) gives a feedback to the user with stress colors in real-time. (e) Based on the user-selected areas, SS analyzes the stress concentrations to provide the user with suggestions for improved designs. (f) The user can iteratively improve the design by repeating analysis-suggestion-choice iterations until a satisfactory result is obtained. (g) The bookshelf is then assembled by the user with metal wires and 3D-printed connectors. (h) The users can also finally attach panels to the connectors for better aesthetics.**

*Both authors contributed equally to this research.



*CHI 2019, May 4–9, 2019, Glasgow, Scotland, UK*

ACM ISBN 978-1-4503-5970-2/19/05…$15.00
https://doi.org/10.1145/3290605.3300893

## ABSTRACT

Current Computer-Aided Design (CAD) tools lack proper support for guiding novice users towards designs ready for fabrication. We propose Shape Structuralizer (SS), an interactive design support system that repurposes surface models into structural constructions using rods and custom 3D-printed joints. Shape Structuralizer embeds a recommendation system that computationally supports the user during design ideation by providing design suggestions on local refinements of the design. This strategy enables novice users to choose designs that both satisfy stress constraints as well as their personal design intent. The interactive guidance enables users to repurpose existing surface mesh models, analyze them in-situ for stress and displacement constraints, add movable joints to increase functionality, and attach a customized appearance. This also empowers novices to fabricate even complex constructs while ensuring structural soundness. We validate the Shape Structuralizer tool with a qualitative user study where we observed that even novice users were able to generate a large number of structurally safe designs for fabrication.



## CCS CONCEPTS

• **Human-centered computing → User interface design**.

## KEYWORDS

Fabrication, CAD, 3D modeling, design recommendation.

**ACM Reference Format:**

Subramanian Chidambaram, Yunbo Zhang, Venkatraghavan Sundararajan, Niklas Elmqvist, and Karthik Ramani. 2019. Shape Structuralizer: Design, Fabrication, and User-driven Iterative Refinement of 3D Mesh Models. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2019), May 4–9, 2019, Glasgow, Scotland, UK*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3290605.3300893


## 1 INTRODUCTION

With the advent of additive manufacturing technology, such as 3D printing and personal fabrication tools, DIY fabrication projects have become increasingly popular. However, fabricating large-scale and functionally sound 3D structures require resources and knowledge that are still not readily available to everyone, making it the exclusive domain of industry professionals. Modern computer-aided design (CAD) applications, sophisticated structural optimization software [36], and fabrication tools are unsuitable for novice users, yet are required for defining general-purpose material removal operations performed by industry-grade mills, lathes, drills, and CNC machines. The same is true for optimization software. Meanwhile, current consumer-level 3D design and fabrication tools are far too simplistic for functional prototypes, due to low fidelity, lack of precision, and a lack of integrated optimization strategies to guide design.

To close this gap, we present Shape Structuralizer, an end-to-end system that allows novice users to design, analyze, and fabricate functional load-bearing structures with minimal expertise. Unlike traditional tools, Shape Structuralizer employs in-situ finite element analysis (FEA) [14], similar to other current design-for-fabrication tools (e.g., TrussFab [21] and FrameFab [17]), but also extends the functionality to help the user locally refine the structures based on their design intent. To the best of our knowledge, our system is the first to incorporate algorithmic analysis of stress distribution within a design-recommendation tool. This approach curbs the need for the user to be trained in structural design by automatically analyzing the current 3D design and providing a ranked list of alternatives for concrete changes that will make the design more structurally safe. Change alternatives are ranked based on how they affect the degree of structural reinforcement and the cost of production. We do not automate these changes to preserve the user's agency in personalizing the product based on their design intent and needs, such as locations where they want to refine the structure, control the degree of increase in complexity, modify the visual appearance, specify the load-bearing capability, or cap the cost of production.

Our work is motivated by a lack of design tools for structural, large-scale, affordable, and rapid fabrication that are accessible to users without special training. In the design process, it is well known that design tools both enable as well as constrain our knowledge and actions [7]. A culmination of advances in both software and hardware has brought novice users closer to full 3D modeling. However, the gap between design and fabrication still requires design tools that are aware of the limitations in fabrication. We contend that with the development of narrow-scope design tools (e.g. [8, 10, 23, 24, 34, 39, 43]) such as ours, users can come even closer to fast structural fabrication of larger constructions, thus letting us democratize processes supporting design-through-fabrication.

In this work, we lower the expertise barrier for users to design and fabricate personalized constructs. We go beyond just supporting the shape of the design to also support in-situ intuitive analysis, influencing the user towards feasible and structural designs that can be fabricated, even ones capable of motion at user-defined locations. Our vision is to empower users without special training to explore the feasible design space while satisfying the user's requirements, i.e., personalized appearance, structural soundness, and movable joints providing added functionality.

To validate Shape Structuralizer, we performed a qualitative user study involving two groups of self identified novice designers, with six designers to each group: one group created designs manually, while the other had access to the design recommendation engine. In our study, we exposed participants to a variety of use cases that demonstrate different capabilities of Shape Structuralizer, such as scalability (Fig. 9), ability to bear load, fidelity (Fig. 11), and articulation of joints (Fig. 8). Our results show that the group with access to the recommendation engine was able to arrive at safer designs faster as well as iterate through a larger number of designs than the group who had no such access.

The primary contribution of SS is the addition of a recommendation system that guides a novice user to make sound structures while preserving their personal design intent. In other words, involving a human in the loop in tandem with the recommendation-based ranking is a primary aspect of the system. It is this hybrid nature that enables efficient exploration of the design space [6] for a 3D model. A fully manual system would require FEA expertise on behalf of the user, whereas a fully automatic one would eliminate the user's personal agency. The contributions of our work are the following:

(1) A novel design-recommendation algorithm that enables a novice user to ideate among a variety of design solutions by analyzing a scaffold structure.
(2) An end-to-end system for novice users, encompassing the work-flow of design, analyze, refine (based on recommendations), and fabricate functional load-bearing structures with minimal expertise.
(3) A human-in-the-loop based design space exploration system, that guides a novice user to make sound structures while preserving their personal design intent, which we validate with a user study.

## 2 RELATED WORK

Shape Structuralizer leverages, inspires, and builds upon a recent body of work in the fabrication area. Here we review the literature in these areas.

### Functional Design and Fabrication

Design and optimization of space structures have a rich history for automated design [13], and for design of elegant structures with relatively simple manufacturing processes for bikes, cars, airplanes, and architectural structures [15]. However, in spite of such a rich history, these methods are inaccessible for common people, especially they do not offer a fully connected package that includes design, analysis and fabrication. Zehnder et al. [41] define the system as a globally coupled energy-minimization problem, discretized with piecewise linear curves that are optimized in the parametric space of a smooth surface. They implement a structural analysis tool that uses eigenanalysis to identify potentially large deformations between geodesically-close curves and guide the user in strengthening the corresponding regions Additionally, they propose 3D printing the structure, thereby imposing print volume restrictions.

Pteromys [39] offers a new work-flow for design and fabrication of optimized glider designs by novice users. Based on this compact aerodynamics model, the design tool supports user-created wing configurations interactively optimized to maximize flightability. StrutModeling [22], allows novice users to prototype 3D models by assembling struts and hub primitives in physical space. The individual struts can be adjusted in length and the angles can be varied by the user using a microcontroller. However, the system has limitations in the maximum length of a strut (170 mm), angle (49.8 degrees), scalability, lacks any means of validating the user's design, and is restricted to only wireframe structures (no panels). TrussFab [21] uses bottles as members and 3D printed connectors for making unit shapes which are then multiplied to obtain large scale structures carrying significant loads, such as human weight. In previous work, the user is expected to understand the FEA results and enhance the model based on their experience and expertise. SS, with its recommendation mechanism, eliminates the need for such expertise. Simply put, the difference between SS and existing approaches (such as TrussFab) is that while previous work in this area lets the user know "where" to reinforce a structure, SS also suggests "how" to reinforce the structure.

### Mixed 3D Fabrication

FrameFab [17] fabricates frame structures using a 6-axis KUKA robotic arm with a customized extrusion head capable of printing self-supporting nodes. Several interactive fabrication strategies offer interactive printing, fast fabrication, and designing while printing. WirePrint [25] prints low fidelity wireframe previews using current 3D printing technology. While their method allows for rapid prototyping, the approach is not viable for creating stable functional prototypes. Other works focus on introducing intermediate low-fidelity fabrication into the traditional high-fidelity but slow 3D printing process, such as printing Wireframe mesh of an object, and substituting sub-volumes of an object with standard Lego blocks [26], or laser cut acrylic panels [5]. A comprehensive review of many such approaches have been provided by Baudisch and Mueller [4]. WireFab [23] supports the designing of such mixed modality (wire bending and laser-cutting) structures in greater detail while taking advantages of 3D-printed joints. The focus of their work is mainly on aesthetic appearance rather than structurality. Also WireFab deals with wire bending, which requires special equipment for fabrication.

### Fabrication-Aware Shape Design

Traditionally, the lack of fabrication knowledge and structurality during design phase results in a number of iterations among the experts in manufacturing and design, even resulting in courses on design for manufacturability. Much of the recent research in fabrication considers manufacturability of the resulting structures during the design stage [31, 40, 42, 43]. In the present design to fabrication process, fabrication of the structure is fairly simple involving use of structural members and joining them to the 3D printed joints. The joint designs are optimized for connectivity and 3D printing. We avoid much of the traditional fabrication problems since we limit it to the joint design

### In-Situ Analysis and Validation in Design

FEA is conventionally used to validate the design by an expert, which usually is not the designer. The design and analysis iteration are separate processes, consume time, and require expertise in both design and analysis, which is a barrier for novice users. Moreover, FEA is predominantly implemented later in the design stage, as it requires high computational power and expertises. Therefore, it is usually ignored during the early design stage. Researchers have attempted to make analysis tools available to novice users in the early design stage, such as FEASY [27], STRAT [29], and 2DSketchFEA [18] leverage 2D sketch interface to enable an analysis in early design stage. Alternately, Shape Structuralizer targets the context of 3D fabrication at design time, which is a more challenging problem.

Other researchers study interactive FEA methods [2, 3, 9]. Their perspective differs from ours as they focus on the computational method for FEA rather than the interactive tools for fabrication. Recent work [32] describes a fast interpolation-based FEA method for design space exploration. A 3D design is parameterized and FEA is precomputed on it. Then, users can iteratively adjust the shape of the design and the analysis results will be updated automatically. This tool is mainly for exploring design space and requires a parameterized template and a precomputed FEA, which is different from our context. A similar work for 3D computational fluid dynamics simulation was explored by Umetani and Bickel [38].

## 3 OVERVIEW

Shape Structuralizer enables users to generate structurally-sound scaffolding constructions from existing 3D models with articulating joints. Our computational design platform workflow unfolds as follows: the user (a) inputs an existing 3D mesh model, (b) decides the scale for the construct (diameter of the rod used and corresponding connector size is generated), (c) chooses to either automatically generate or interactively define scaffolding structures, (d) performs an in-situ structurality analysis, and (e) edits the scaffolding structures based on a novel recommendation system. Steps (d) and (e) can be iteratively run multiple times until the resultant scaffolding structure simultaneously satisfies both the user's aesthetic and structurality requirements. Users can edit the scaffolding system by either changing the shape of the scaffolding structure or adding/deleting vertices and edges of the scaffolded structure. Below we discuss each parts of this interactive pipeline in detail.

## 4 COMPUTATIONAL DESIGN FRAMEWORK

A scaffolded structure modeled from a mesh model consists of "edges," which are straight metal wires linked together by means of 3D printed connectors ("vertices"). Our geometric modeling tool allows the user to efficiently define and edit vertices and edges through simple and direct interactions.

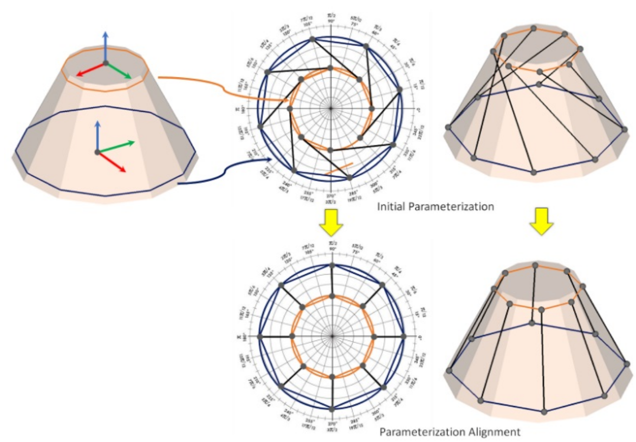


**Figure 2: For two neighboring cross-sections, independent parameterizations in different local coordinates systems lead to correspondence of vertices with large twisting (a). By aligning the local coordinates of two cross-sections and re-parameterizing the two cross-sections, the correspondence of vertices show small twisting (b).**

*Cross-Section Creation and Editing.* To quickly obtain a rough scaffolding shape, planar cross-section extraction tool is provided by our platform. A plane specified by user's stroke is used to cut the 3D mesh model and a resultant cross-section contour curve is obtained. Contour curves are chosen for generating scaffolds as they efficiently capture the input model's shape. The generated contour curves cannot directly be used to form the scaffolding structure as the unnecessary shape details will cause difficulties during fabrication. Thus, a curve approximation is performed to simplify a contour curve while ensuring that the simplified curve approximates its original shape. For each contour curve consisting of $M$

points and having total length $L$, an arc length parameterization is conducted [12] to map all the points onto a circle centered at original with $\rho = \frac{2\pi}{L}$ as radius, and then $N$ points are uniformly sampled in parametric domain ($N \ll M$). In all the tests, the value of $N$ has been set to 4 or 8, which balances the shape approximation and complexity for fabrication and assembly appropriately.

Rather than creating individual cross-sections and linking them manually, we use an automatic cross-section linking function to connect two neighboring cross-sections to form a frustum structure. When the user interactively defines a cross-section $C_i$, the tool automatically detects its previously created cross-section $C_{i-1}$, and links $C_i$ and $C_{i-1}$ at the corresponding vertices. For each vertex $\mathbf{V}_j^{C_i}$ on $C_i$, a corresponding vertex $\mathbf{V}_j^{C_{i-1}}$ on $C_{i-1}$ is the one with the same parameter. As the parameterizations of $C_i$ and $C_{i-1}$ are independent, the corresponding vertices have multiple solutions. To form a frustum structure, it is desired that the corresponding two vertices have minimal twisting. Therefore, an optimization algorithm is implemented to identify the correspondence of vertices with the least twisting. First, local coordinates systems are established on each contour curve, and the arc-length parameterizations are conducted on each contour independently. Then all the local coordinate systems of a contour are aligned to a reference coordinate system by computing a minimum rotation from all other local coordinate systems to the reference one, and then the parameterization is recomputed. Finally, the cross-sections are uniformly sampled in a parametric domain and the corresponding vertices are linked together to form a frustum structure. The reference coordinates system could be either fixed to the first cross-section, or updated to the previous cross-section. In our experiments, we observed that fixing the reference cross-section to the first one generates less twisting in most cases.

*Auto-Scaffolding.* An auto-scaffolding tool is provided to the user for automatically generating an initial scaffolding structure. The user can keep on editing the initial structure using other interactive tools until it is satisfactory. Following Garland and Heckbert [16], our auto-scaffolding allows the user to specify a target vertex number (usually much smaller than the number of vertex of the original model), and control the shape approximation of the simplified structure to the original model by minimizing the quadric error metrics. We also enforce two fabrication restrictions in the auto-scaffolding: 1) the wire length cannot be smaller than 3 cm, 2) and the angle between two wires cannot be less than 35 degrees. Fig. 4 shows a result generated by the auto-scaffolding tool on the dog head model. Together with the interactive tools, the user are able to generate the customized wire structures efficiently.

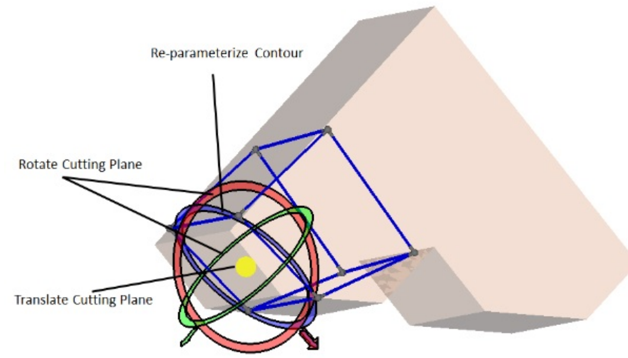


**Figure 3: Widget-based interface tool to edit the shape of a selected cross-section. The red and green markers indicate rotating the cutting plane to generate different contour curves, while the blue one is used for adjusting the parameterization of the cross-section on the same contour curves. The yellow sphere is the center of the widget and is used for translation.**

*Editing.* If the initial cross-section is not satisfactory, a widget-based tool (Fig. 3) can be used to manipulate the shape of the cross-section as the user desires. The two widgets perpendicular to the cutting plane can be used to adjust the orientation of the cutting plane for different contour curves. The newly generated contour curves are simplified to cross-sections in real time and the twisting minimization is also enforced here. The other widget on the cutting plane does not change the contour curves. Instead, when the user rotates the widget, the contour curve will be rotated in the parametric domain and a re-parameterization will be conducted as well. The positions of vertices on the cross-section will be updated accordingly. The user can also translate the cutting plane by selecting and dragging the sphere centered in the widgets.

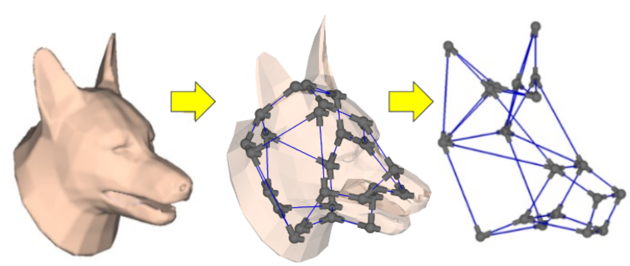

**Figure 4: A dog head model scaffolded using Shape Structuralizer's auto-scaffolding tool (left to middle). More detailed shapes can be obtained by creating/deleting vertices and edges (right).**

*Vertex Edge Creation and Deletion.* Cross-section creation and editing tools enable the user to create a roughly scaffolded structure efficiently. This tool limits the user's ability to generating frustum-like structures. To enhance the ability of modeling other shapes, we also provide the user a set of less-efficient but more flexible interactive tools to create vertices on the surface of the model and link two vertices to create an edge. Users additionally have the option to delete vertices or edges. These tools provide the user the ability to define and manipulate basic elements of the scaffold structures. We expect the user use these tools to create detailed shapes after defining a rough shape using cross-section based tools or the auto-scaffolding tool (Fig. 4).

### Structural Analysis

Most objects in everyday life require load bearing. Hence, in order to facilitate users to produce personalized scaffolds that are also load-bearing, our software embeds a tool that provides feedback and displays a dashboard with the elements that are most likely to fail. Thus, the user are capable of taking informed decisions to modify the design.

*Supporting User Decision during Design.* We scaffold structural stability by incorporating an on-demand, user-driven Finite Element-based Analysis (FEA) during the user' s design. We choose a mechanics-based FEA model by treating each member as a spatial frame element. A frame element is a superposition of a beam and truss element. Hence, the analysis can handle bending moments, transverse, and axial loads [19] [20]. Shape Structuralizer handles the analysis on its own and does not depend on any external FEA engine, which allows providing real-time feedback. Additionally, the analysis module within our control provides greater flexibility to control visualization for designing visual feedback and reduces the cognitive load on users while building an end-to-end application.

*Interactive Tools for Structural Analysis.* The user interaction capabilities support tools to add loads and boundary conditions (BCs) through simple interactions using a widget-based interface. Users can define and modify both the direction and magnitude of the forces applied on selected vertices using a widget-based interface. To ease the task of setting BCs for novice users, they can simply select the nodes that are in contact with the ground. The degrees of freedom appropriate for the context are automatically constrained. Within this context, users no longer require knowledge of the FEA-based degree-of-freedom boundary conditions. The more experienced users are also given the additional ability to modify these preset constraints to represent a more accurate boundary condition for special cases. Currently, the loading is limited to the connectors. Even if personalized panels using cardboard are implemented, the loading from the panels is transferred to the underlying connectors. Therefore, loads have been reduced into point loads at each of the connector nodes. Here, an implicit assumption is that the load is not shared by the panels and results in a conservative design scenario.

We have validated solutions from Shape Structuralizer against the standard numerical solver Abaqus [1] under the same external conditions. We observed high consistency across results.

## 5 RECOMMENDATION SYSTEM

After a preliminary structural analysis of the design through FEA, we compute and display to users the stress at each vertex through red-blue color codes (red for high stress, blue for low). According to continuum mechanics, the creation of continuity near regions of high stresses reinforces the structure. Therefore, in design practice, regions of high stress are usually refined to reinforce the structure. A simple and effective way to reinforce wire structure is to add more vertices and more edges around high stress regions. Determining where the vertices should be located relies on structure analysis expertise, which is not present in novice users. Furthermore, there are fabrication constraints on the newly added vertices, which brings more difficulty to novice users. The recommendation system assists novice users to reinforce their design by automatically suggesting different refinement strategies, ranking these strategies based on stress, and enforcing the fabrication requirement.

### Subsection Analysis

Our subsection analysis method is inspired by a well-studied FEA modeling technique called sub-modeling [30]. Sub modeling, proposed in the 1970s, refines the mesh density of the region of interest and performs analysis only in the refined region, in order to save computational power and achieve high computational efficiency. The mesh becomes denser in the region of interest, and the boundary conditions from the former coarse analysis are inherited. Borrowing the idea of the localized refinement and analysis, our subsection analysis will be performed in a single user-specified region. With the color displayed at each vertex on the initial structure (see Fig. 5 (a)), the user is able to select vertices with large stress values. Then, the selected vertex and its neighboring regions are defined as region of interest for the following analysis. In order to achieve high performance with interactive feedback, we only include the selected vertex $\mathbf{V}_i$, and its two neighboring vertices $\mathbf{V}_j$ and $\mathbf{V}_k$ with largest stress values among $\mathbf{V}_i$'s neighbors (shown in Fig. 5 (a)).

### Refinement and Stress-aware Ranking

The selected region of interest introduces an essential question: ***What changes are required to improve load carrying capability?*** One approach is the topology optimization (TO) method [33], which generates an optimal structure based on the given boundary conditions. However, TO method is heavily time consuming and fails to achieve real-time performance. Conversely, an optimal solution is not always necessary for the given system since design is often open-ended due to a variety of objective functions and solutions. Therefore, rather than generating an optimal structure, a few refinement strategies are employed based on experienced designers' heuristics to explore the possible design space. Specifically, three strategies are adopted.

**Type1**: Inserting one vertex $\mathbf{V}_{ij}$ between $\mathbf{V}_i$ and $\mathbf{V}_j$, and another vertex $\mathbf{V}_{ik}$ between $\mathbf{V}_i$ and $\mathbf{V}_k$. Adding vertex $\mathbf{V}_c$ at the intersection of line $\mathbf{V}_{ij}\mathbf{V}_k$ and line $\mathbf{V}_{ik}\mathbf{V}_j$, and link $\mathbf{V}_c\mathbf{V}_i$, $\mathbf{V}_c\mathbf{V}_j$, $\mathbf{V}_c\mathbf{V}_k$, $\mathbf{V}_c\mathbf{V}_{ij}$, and $\mathbf{V}_c\mathbf{V}_{ik}$ accordingly, in order to fulfill fabrication constraints (see top of Fig.5 (b)).

**Type2**: Inserting one vertex $\mathbf{V}_c$ at the centroid of $\Delta\mathbf{V}_i\mathbf{V}_j\mathbf{V}_k$, and link $\mathbf{V}_c\mathbf{V}_i$ (see middle of Fig. 5 (b)).

**Type3**: Simply link the mid point of $\mathbf{V}_i\mathbf{V}_j$ and the mid point of $\mathbf{V}_i\mathbf{V}_k$ (see bottom of Fig. 5 (b)).

For **Type1** refinement, the position of the $\mathbf{V}_{ij}$ is determined by this equation $\mathbf{V}_{ij} = \mathbf{V}_i + \alpha(\mathbf{V}_j - \mathbf{V}_i)$. The value of $\alpha$ is computed through $\frac{S_j}{S_j+S_i}$, where $S_i$ and $S_j$ are the stress values on $\mathbf{V}_i$ and $\mathbf{V}_j$. By using this ratio, $\mathbf{V}_{ij}$ will be inserted at close proximity to the vertex with higher stress. This reiterates the principle of reinforcing the locations with higher stress concentration for better stress dispersion.

The three refined sub-structures are analyzed and ranked according to the average stress on them. Again, the subsection analysis ensures computational efficiency and provides real-time feedback. This analysis-refinement process is iteratively performed by users until one of two conditions is achieved: 1) no nodal stresses are greater than the yield stress of the material; or 2) the distance between two neighboring nodes is less than 3 cm, as it introduces complexity during construction. There might be a case where a system fails to reach safety even after multiple refinements and might exit from the iterative loop upon hitting the second constraint. It is noted that this case might not be due to the limitation of the system itself, but rather due to the limiting factor of the nature of the material and loading conditions.

All three reinforcement strategies utilize the triangle principle, which states "a triangle loaded at its corners is the most efficient 2D structure" [35]. Stress flows from a region of high to lower stress concentration. The SS strategies are all based on allowing more pathways between two such regions, which improves stress distribution and rigidity.

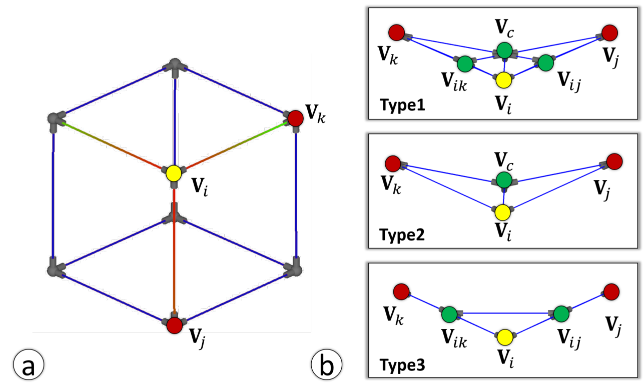


**Figure 5: The Users select a vertex $\mathbf{V}_i$, and $\mathbf{V}_i$'s two neighbors $\mathbf{V}_j$ and $\mathbf{V}_k$ with highest stress are retrieved by our system (a). The three types of refinement strategies are shown in (b).**

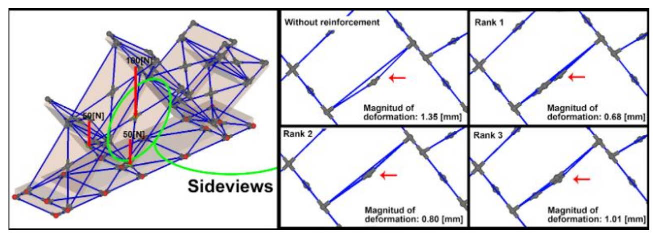


**Figure 6: Ranking algorithm results. Ranking algorithm validated by minimum deformation.**

| Nodes | Matrix Size | Using SSA (secs) | Without SSA (secs) |
|---|---|---|---|
| 8 | 2,304 | 0.021 | 0.033 |
| 27 | 26,244 | 0.234 | 0.421 |
| 64 | 147,456 | 1.314 | 3.380 |
| 125 | 562,500 | 4.822 | 10.610 |

**Table 1: Computation time with and without sub-section analysis (SSA).**

### Validation

The above image shows the different rates of deformations experienced by a subsection of a book case, as suggested and predicted by our ranking algorithm. The deformations in the real world (Fig. 7) are small, hence not visible without specialized equipment to measure them. The deformations from the model (Fig. 6) have been uniformly scaled for exaggeration purpose.

Table 1 describes the computation time comparison with and without sub-section analysis. Sub-section analysis leads to reduced time in computation, which in turn leads to a better user experience.

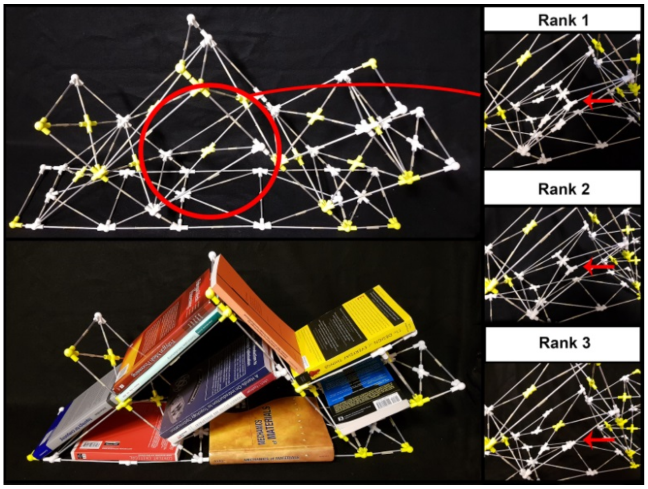


**Figure 7: Fabrication of the physical bookshelf previously analyzed. We leveraged the capabilities of the recommendation tool in creating this bookshelf.**

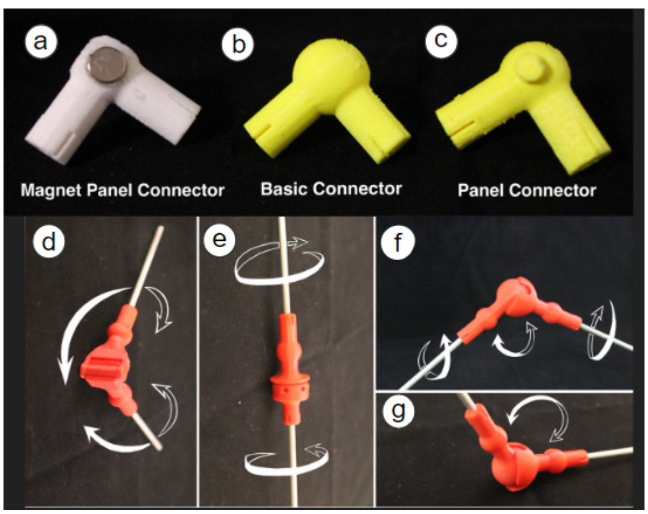


**Figure 8: From left to right, there are three types of connectors: a) magnet panel connector b) basic connectors, c) panel connectors for wires and attaching 3D printed panels. Fig d) to g) shows the different types of articulating joints.**

## Connector Design

*Rod-Element Connection.* The Connector design features two main components. The central component, a sphere, is used to connect multiple hollow cylindrical geometries which house the rod element. These cylindrical components have an internal step that allows for a transition fit .

The fabrication of the connectors was performed using MOJO Desktop printer [37] (FDM Printer) by Stratasys. It should be noted that fabrication by 3D printing (ABS) has several drawbacks with respect to accuracy. As mentioned by Dimitrov et al. [11], the factors for the loss in accuracy can be attributed to nature of material used, nominal dimensions, build orientation, geometric features and their topology, wall thickness, post treatment procedures, and infiltrating agent. Therefore, several iterations were performed with the same printer until the connector maintained a transition fit with the wire element.

*Panel-Wire Element Connection.* Personalization with respect to appearance in shape structuralizer is enhanced by controlling the fidelity of the wire frame model. The user has the option to select a low fidelity model which only consists of wire elements and the connectors. Additionally, for a higher level of detail, the user may decide to add appearance using cardboard as panels.

*Articulation Joints Connection.* Shape Structuralizer additionally gives the user a library of movable joints to add to their scaffolded construct. Primarily, the joints provide the user' s model additional degrees of freedom increasing the functionality of their structure.

The users were provided with a ball and socket-joint connector that gave the user three rotational degrees of freedom to certain parts of their scaffolded structure. This setup allowed the user to generate scaffold structures capable of rotating and tilting motions. A hinge joint can be implemented by the users to provide a single degree of freedom by providing motion in one plane. Such component empowers the scaffolded structure to tilt without rotation. The bearing joint also provides only a single degree of freedom, rotation, about any one axis. The user could incorporate rotating mechanisms without tilting using this joint.

*Scalability of Connectors.* Shape Structuralizer allowed the user to decide the scale of their construct. They were able to choose between a range of rod diameters. In our work, we have showcased two different dimensions of 3.175 mm and 8 mm diameter rods. However, the software is capable of generating a variety of connectors to accommodate different rod diameters. Using larger diameter rods improves the structural properties of the system. It allows the model to handle more stress and empowers the user to create larger structures. Additionally, the user may choose to combine the smaller and larger scale modes to create large structures for the main body using larger connectors and add detail using the smaller connectors for certain sections of the model.

*Fabrication and Assembly Guidance.* Upon completing a satisfactory scaffolded structure, the user interface provided capabilities to save the wireframe files. The interface allotted the user to generate the STL files for the connector, which are rendered using OpenSCAD [28], with the help of scripts generated by Shape Structuralizer. Then, it was printed using standard 3D printers. To ease the assembly process, Shape

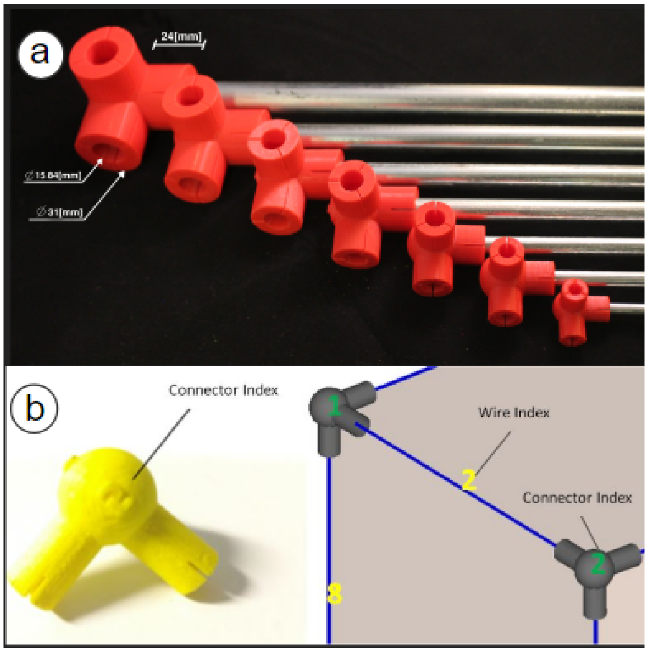


Figure 9: a) Shape Structuralizer allows users to use metal wires or rods with a diameter ranging from 3.175 mm to 15.84 mm. b) SS displays indices of both connector and wires as a reference for the user to assemble them. The corresponding indices are also embossed on connectors, and attached on wires.

Structuralizer displays the connector and wire index on the interface. Also, it generates a wire length file that lists wire indices and their corresponding lengths. The connectors were also numbered to assist the user during assembly.

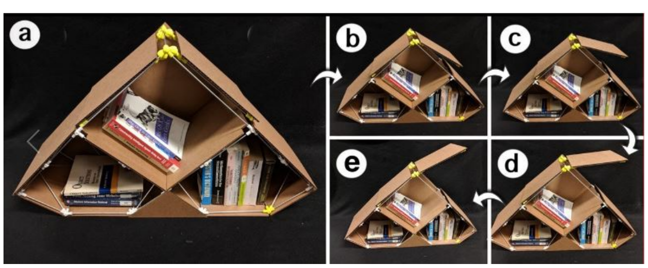


Figure 10: A bookshelf successfully takes loads of several books, and has a movable door enable by articulating joints.

## 6 EVALUATION

We have evaluated the functional performance, and conducted a qualitative user study of Shape Structuralizer.

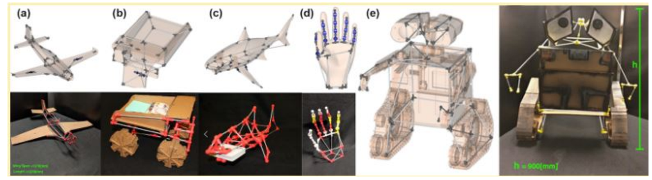


Figure 11: Prototype results generated by Shape Structuralizer (left to right): (a) an airplane ($1,270 \times 1,270$ mm), (b) dump truck, (c) 3D shark card holder, (d) hand 3D model with movable joints, (e) and a WallE designed with movable joints, and fabricated in 1:1 size at 900 mm.

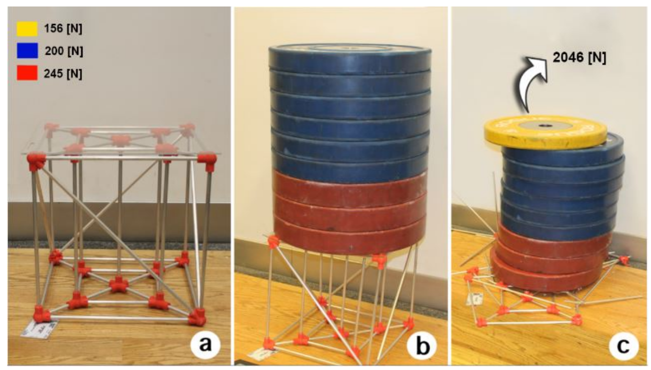


Figure 12: This cube with 8 mm rods can bear up to 2046 N.

### Use Cases

Here, in order to validate each Shape Structuralizer capability, including scalability, ability to bear load, fidelity, and articulation of joints, we demonstrate several examples of corresponding physical structures.

*Scalability.* Shape Structuralizer supports design and fabrication of objects in various scales. The family of connectors allows the use of wires and rods with different diameters (from 3.175 mm to 15.84 mm); therefore, smaller objects, such as a hand, and larger objects, such as WallE can be designed and fabricated using our system.

*Fidelity.* The modeling capability of our system enables the design of high fidelity shapes. As shown in Fig. 11 (b), the key geometric features of the dog model (Fig. 4) and the shark model (Fig. 11) are well-captured by the users using our modeling tools. Also, users have complete control on the level of fidelity by manipulating the position of points interactively.

*Load-Bearing Ability.* Shape Structuralizer's recommendation engine substantially improves the structurality of the objects. As shown in Fig. 12, a cube structure with a length of a foot per each side is modeled through the design-optimization iterations. The fabricated structure was able to handle 430 lbs (Red = 55 lbs, Blue = 45 lbs), and collapsed upon further addition of weight. This calculation concurs with the results of our tool. Thus, the tool can produce models sustainable against heavy loads and has the potential to develop personalized furniture.

*Articulation of Joints.* Shape Structuralizer supports additional functionalities by allowing users to interactively specify articulation joints. Different joints with degree-of-freedoms are added onto the designed shapes, including the hinge joints for bookshelf, airplane, shark and WallE, the rotary joints for the dump truck, and the ball joints for the hand. The interaction tool and the family of articulation joints enables novice users to design and fabricate objects with motion functionalities.

## User Study

We evaluated the utility and user experience of Shape Structuralizer through a qualitative user study.

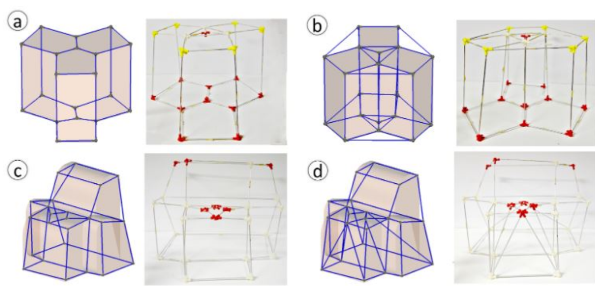


**Figure 13: Sample structures created from the user study.**

We recruited a total of 12 participants for the study. The users selected for the study were self-identified as novices in the field of design and fabrication, and none of them had any prior experience in using this system. The users where split into two groups. **Group 1** was a control group with no access to the recommendation tool, whereas **Group 2** has full access to SS's recommendation engine.

The users were first given a tutorial of the interface and were allowed to accustom themselves to the software package. Then, they were given a physical copy of a connector and the wire element that they would be using throughout the study. Each user was tasked with scaffolding and re-purposing two models: (1) a cube of dimensions $15x15x15$ $cm^3$, and (2) a surface model of a functional object such as a book-shelf or table or a chair. Additionally, to provide spatial cognition, they were provided with a measuring tape, the weight that they would be using during the course of the study, a physical sample of a wire frame geometry, and a 2D diagram of the user study model for details regarding the various dimensions.

# 7 RESULTS AND DISCUSSION

Here we discuss our findings in more detail.

## Study Results

Overall our results show that, Group 2—who had access to the full SS interface—were able to construct structurally safe and load-bearing designs much faster than Group 1, who could only use basic 3D modeling functionality. What's more, Group 1 also generated a much larger number of designs throughout this process. This, in particular, is a significant finding given that high-volume ideation is known to be correlated with high quality of the final design.

The users were first tasked with scaffolding a cube surface and were tasked to design it such that it can carry approximately 2 books (6 kg). This was done to make all the users familiar with the UI. It was observed that the time taken by the user to complete this part of the study ranged between 4 to 6 mins. All users where able to achieve a structure capable of handling the load of 6 kg.

Subsequently, they were given the task of scaffolding either a table, chair, or a bookshelf. In each case they were given constraints on the load that each system would take 6 kg. The range of time taken by the individuals in Group 1 without the recommendation tool ranged between 15 to 35 min, with multiple iterations, except one exception who completed the task in 11 min. The most noteworthy observation with Group 2 was that all designers completed their goal in under 10 min. The max time was recorded for this group was 9.42 min. Every user in Group 2 made use of the recommendation system.

The difference in time between the users can be attributed to the fact that the recommendation tool effectively reduces the time taken for cognitive processing for structural refinement. Furthermore, the recommendation tool directly creates the nodes and its corresponding elements with a single click of a button. This functionality saved precious time otherwise spent in manually creating nodes and elements by the control group. Finally, it was to be noted that all the users who took part in the study where able to achieve their target by creating a structure that could handle the required load.

It was noted that, in some cases, the users were so focused on the functional attribute of the design, that they neglected the appearance of the object being designed. We speculate that this arose due to increased cognitive load on the user as their focus shifts from focusing on the personalization and appearance and more towards satisfying the functional requirements. The reinforcing ideology used by the users relied on the designs they had observed in everyday life, such as trusses from bridges or mesh pattern in shopping carts.

### Limitations

In our current implementation, all the wire elements were hand-cut, which lead to dimensional inaccuracies that stacked. Coupled with 3D printing inconsistencies, this lead to an uneven stress flow in the final assemblies. To address this issue, we can also incorporate dimensionality variation and assembly feedback, to incorporate such variations into tolerances of individually fabricated components.

Based on informal feedback during our experiment, we learnt that significant user interface improvements could be made. Unobtrusively providing design recommendations in an interactive interface is a challenging proposition, and one we could do better.

SS is limited to bear tensile load, but the kind of load (tensile or compressive) will not diminish the user's experience to design better structures based on the recommendation system, as the design principles behind both remain the same.

There are a few fabrication limitations such as: (a) the size of any two nodes must be greater than 3 cm, and (b) the angle between two wire elements must be greater than 35 degrees for ease of assembly. New nodes cannot be created that violate these two constraints.

## 8 CONCLUSION AND FUTURE WORK

We have presented Shape Structuralizer, an end-to-end computational design tool that is capable of design, analysis, iterative recommendation-based refinement, and fabrication of functional load-bearing structures with minimal expertise. We developed a powerful set of user-driven design and editing tools for fabricating structurally sound scaffolds. We also provide a novel design recommendation system that has been proven to enhance the design capability of novice users. As evidenced by the simulation and fabrication of design through our system, wire-scaffolds are well suited to represent a large array of functional real-world objects.

In the future, we plan to improve UI functionalities by adding symmetric creation tools, and tools to create interior vertices. In our use cases, cardboard panels are attached onto the scaffolding structure. 3D printed panels can also be generated and attached to the scaffolded structure for higher aesthetic capability. Additionally support for tensile loads can be accommodated by modifying the connector design such as adding threads to the connectors or a locking features.

## ACKNOWLEDGMENTS

We would like to thank Prof. Jack Hu and Prof. Thomas Siegmund for their valuable feedback. We also thank Javier Belmonte, Aditya Dantam, Sinuo Suo, and Theerakorn Prasutchai for their help with fabricating and assembling the use cases. We would also like to thank Ana Villanueva for her help with the user studies.

This work was supported by U.S. National Science Foundation awards IIS-1422341 (http://www.nsf.org/) and EAGER-1547134 (http://www.nsf.org/). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agency.

## REFERENCES

[1] Abaqus 2018. Abaqus Unified FEA Complete Solutions for Realistic Simulation. (2018). Retrieved January 6, 2019 from https://www.3ds.com/products-services/simulia/products/abaqus/abaquscae/

[2] Jernej Barbič, Marco da Silva, and Jovan Popović. 2009. Deformable object animation using reduced optimal control. *ACM Transactions on Graphics* 28, 3, Article 53 (2009). DOI: http://dx.doi.org/10.1145/1531326.1531359

[3] Jernej Barbič, Funshing Sin, and Eitan Grinspun. 2012. Interactive editing of deformable simulations. *ACM Transactions on Graphics* 31, 4, Article 70 (2012). DOI: http://dx.doi.org/10.1145/2185520.2185566

[4] Patrick Baudisch and Stefanie Mueller. 2017. Personal fabrication. *Foundations and Trends® in Human–Computer Interaction* 10, 3–4 (2017), 165–293.

[5] Dustin Beyer, Serafima Gurevich, Stefanie Mueller, Hsiang-Ting Chen, and Patrick Baudisch. 2015. Platener: Low-fidelity fabrication of 3D objects by substituting 3D print with laser-cut plates. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, 1799–1806. DOI: http://dx.doi.org/10.1145/2702123.2702225

[6] Margaret A Boden. 1996. *Dimensions of Creativity.* MIT Press.

[7] Senthil K. Chandrasegaran, Karthik Ramani, Ram D. Sriram, Imré Horváth, Alain Bernard, Ramy F. Harik, and Wei Gao. 2013. The evolution, challenges, and future of knowledge representation in product design systems. *Computer-Aided Design* 45, 2 (2013), 204–228. DOI: http://dx.doi.org/10.1016/j.cad.2012.08.006

[8] Desai Chen, Pitchaya Sitthi-amorn, Justin T. Lan, and Wojciech Matusik. 2013. Computing and fabricating multiplanar models. *Computer Graphics Forum* 32, 2pt3 (2013), 305–315. DOI: http://dx.doi.org/10.1111/cgf.12050

[9] Xiang Chen, Changxi Zheng, and Kun Zhou. 2017. Example-based subspace stress analysis for interactive shape design. *IEEE Transactions on Visualization & Computer Graphics* 23, 10 (2017), 2314–2327. DOI: http://dx.doi.org/10.1109/TVCG.2016.2618875

[10] Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W Sumner, Wojciech Matusik, and Bernd Bickel. 2013. Computational design of mechanical characters. *ACM Transactions on Graphics* 32, 4 (2013), 83. DOI: http://dx.doi.org/10.1145/2461912.2461953

[11] Dimitar Dimitrov, W Van Wijck, Kristiaan Schreve, and Neal De Beer. 2006. Investigating the achievable accuracy of three dimensional printing. *Rapid Prototyping Journal* 12, 1 (2006), 42–52. DOI: http://dx.doi.org/10.1108/13552540610637264

[12] Manfredo P. Do Carmo, Gerd Fischer, Ulrich Pinkall, and Helmut Reckziegel. 2017. Differential Geometry. In *Mathematical Models.* Springer, 155–180.

[13] W Dorn. 1964. Automatic design of optimal structures. *J. de Mecanique* 3 (1964), 25–52.

[14] Jacob Fish and Ted Belytschko. 2007. *A First Course in Finite Elements.* Vol. 1. John Wiley & Sons New York.

[15] Robert M. Freund. 2004. *Truss design and convex optimization.* Technical Report. Massachusetts Institute of Technology.

[16] Michael Garland and Paul S. Heckbert. 1997. Surface Simplification Using Quadric Error Metrics. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques*. ACM, New York, NY, USA, 209–216. DOI: http://dx.doi.org/10.1145/258734.258849

[17] Yijiang Huang, Juyong Zhang, Xin Hu, Guoxian Song, Zhongyuan Liu, Lei Yu, and Ligang Liu. 2016. Framefab: Robotic fabrication of frame shapes. *ACM Transactions on Graphics* 35, 6, Article 224 (2016). DOI: http://dx.doi.org/10.1145/2980179.2982401

[18] Tara C. Hutchinson, Falko Kuester, and Mark E. Phair. 2007. Sketching finite-element models within a unified two-dimensional framework. *Journal of Computing in Civil Engineering* 21, 3 (2007), 175–186. DOI: http://dx.doi.org/10.1061/(ASCE)0887-3801(2007)21:3(175)

[19] D. J N Reddy. 2005. *An Introduction to the Finite Element Method*. McGraw-Hill Education. https://books.google.com/books?id=8gqnRwAACAAJ

[20] Peter I Kattan. 2010. *MATLAB guide to finite elements: an interactive approach*. Springer Science & Business Media.

[21] Robert Kovacs, Anna Seufert, Ludwig Wall, Hsiang-Ting Chen, Florian Meinel, Willi Müller, Sijing You, Maximilian Brehm, Jonathan Striebel, Yannis Kommana, and others. 2017. Trussfab: Fabricating sturdy large-scale structures on desktop 3D printers. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, 2606–2616. DOI: http://dx.doi.org/10.1145/3025453.3026016

[22] Danny Leen, Raf Ramakers, and Kris Luyten. 2017. StrutModeling: A Low-Fidelity Construction Kit to Iteratively Model, Test, and Adapt 3D Objects. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, 471–479. DOI: http://dx.doi.org/10.1145/3126594.3126643

[23] Min Liu, Yunbo Zhang, Jing Bai, Yuanzhi Cao, Jeffrey M Alperovich, and Karthik Ramani. 2017. WireFab: Mix-Dimensional Modeling and Fabrication for 3D Mesh Models. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, 965–976. DOI: http://dx.doi.org/10.1145/3025453.3025619

[24] James McCrae, Nobuyuki Umetani, and Karan Singh. 2014. FlatFitFab: interactive modeling with planar sections. In *Proceedings of the ACM Symposium on User Interface Software and Technology*. ACM, 13–22. DOI: http://dx.doi.org/10.1145/2642918.2647388

[25] Stefanie Mueller, Sangha Im, Serafima Gurevich, Alexander Teibrich, Lisa Pfisterer, François Guimbretière, and Patrick Baudisch. 2014. WirePrint: 3D printed previews for fast prototyping. In *Proceedings of the ACM Symposium on User Interface Software and Technology*. ACM, 273–280. DOI: http://dx.doi.org/10.1145/2642918.2647359

[26] Stefanie Mueller, Tobias Mohr, Kerstin Guenther, Johannes Frohnhofen, and Patrick Baudisch. 2014. faBrickation: fast 3D printing of functional objects by integrating construction kit building blocks. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, 3827–3834. DOI: http://dx.doi.org/10.1145/2556288.2557005

[27] Sundar Murugappan and Karthik Ramani. 2009. FEAsy: a sketch-based interface integrating structural analysis in early design. In *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 743–752. DOI: http://dx.doi.org/10.1115/DETC2009-87727

[28] OpenSCAD 2018. The Programmers Solid 3D CAD Modeller. (2018). Retrieved September 21, 2018 from http://www.openscad.org/

[29] Joshua M. Peschel and Tracy Anne Hammond. 2008. STRAT: a Sketched-truss Recognition and Analysis Tool. In *DMS*. 282–287.

[30] Adhémar Jean Claude Barré de Saint-Venant. 1856. Memoire sur la Torsion des Prismes. *Mem. Divers Savants* 14 (1856), 233–560.

[31] Ryan Schmidt and Nobuyuki Umetani. 2014. Branching support structures for 3D printing. In *ACM SIGGRAPH 2014 Studio*. ACM, 9. DOI: http://dx.doi.org/10.1145/2619195.2656293

[32] Adriana Schulz, Jie Xu, Bo Zhu, Changxi Zheng, Eitan Grinspun, and Wojciech Matusik. 2017. Interactive design space exploration and optimization for CAD models. *ACM Transactions on Graphics* 36, 4 (2017), 157. DOI: http://dx.doi.org/10.1145/3072959.3073688

[33] Ole Sigmund. 2001. A 99-line topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization* 21, 2 (2001), 120–127. DOI: http://dx.doi.org/10.1007/s001580050176

[34] Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross. 2013. Computational design of actuated deformable characters. *ACM Transactions on Graphics* 32, 4 (2013), 82. DOI: http://dx.doi.org/10.1145/2461912.2461979

[35] Stephen P. Timoshenko and Donovan Harold Young. 1945. *Theory of Structures*. McGraw-Hill.

[36] Tosca 2018. Efficient optimization based on FEA and CFD simulations. (2018). Retrieved September 21, 2018 from https://www.3ds.com/products-services/simulia/products/tosca/

[37] TUG 2018. MOJO. (2018). Retrieved September 21, 2018 from http://www.stratasys.com/3d-printers/mojo

[38] Nobuyuki Umetani and Bernd Bickel. 2018. Learning three-dimensional flow for interactive aerodynamic design. *ACM Transactions on Graphics* 37, 4 (2018), 89. DOI: http://dx.doi.org/10.1145/3197517.3201325

[39] Nobuyuki Umetani, Yuki Koyama, Ryan Schmidt, and Takeo Igarashi. 2014. Pteromys: interactive design and optimization of free-formed free-flight model airplanes. *ACM Transactions on Graphics* 33, 4 (2014), 65. DOI: http://dx.doi.org/10.1145/2601097.2601129

[40] Nobuyuki Umetani and Ryan Schmidt. 2013. Cross-sectional structural analysis for 3D printing optimization. In *SIGGRAPH Asia Technical Briefs*. Citeseer, 5–1.

[41] Jonas Zehnder, Stelian Coros, and Bernhard Thomaszewski. 2016. Designing structurally-sound ornamental curve networks. *ACM Transactions on Graphics* 35, 4 (2016), 99. DOI: http://dx.doi.org/10.1145/2897824.2925888

[42] Xiaoting Zhang, Xinyi Le, Athina Panotopoulou, Emily Whiting, and Charlie C. L. Wang. 2015. Perceptual models of preference in 3d printing direction. *ACM Transactions on Graphics* 34, 6 (2015), 215. DOI: http://dx.doi.org/10.1145/2816795.2818121

[43] Yunbo Zhang, Wei Gao, Luis Paredes, and Karthik Ramani. 2016. Cardboardizer: Creatively customize, articulate and fold 3d mesh models. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, 897–907. DOI: http://dx.doi.org/10.1145/2858036.2858362