Volumetric Michell Trusses for Parametric Design & Fabrication

Rahul Arora
University of Toronto
Toronto, ON, Canada
arorar@dgp.toronto.edu

Alec Jacobson
University of Toronto
Toronto, ON, Canada
jacobson@cs.toronto.edu

Timothy R. Langlois
Adobe Research
Seattle, WA, USA
tlangloi@adobe.com

Yijiang Huang
Massachusetts Institute of Technology
Cambridge, MA, USA
yijiangh@mit.edu

Caitlin Mueller
Massachusetts Institute of Technology
Cambridge, MA, USA
caitlinm@mit.edu

Wojciech Matusik
MIT CSAIL
Cambridge, MA, USA
wojciech@csail.mit.edu

Ariel Shamir
The Interdisciplinary Center
Hertsliya, Israel
arik@idc.ac.il

Karan Singh
University of Toronto
Toronto, ON, Canada
karan@dgp.toronto.edu

David I.W. Levin
University of Toronto
Toronto, ON, Canada
diwlevin@cs.toronto.edu

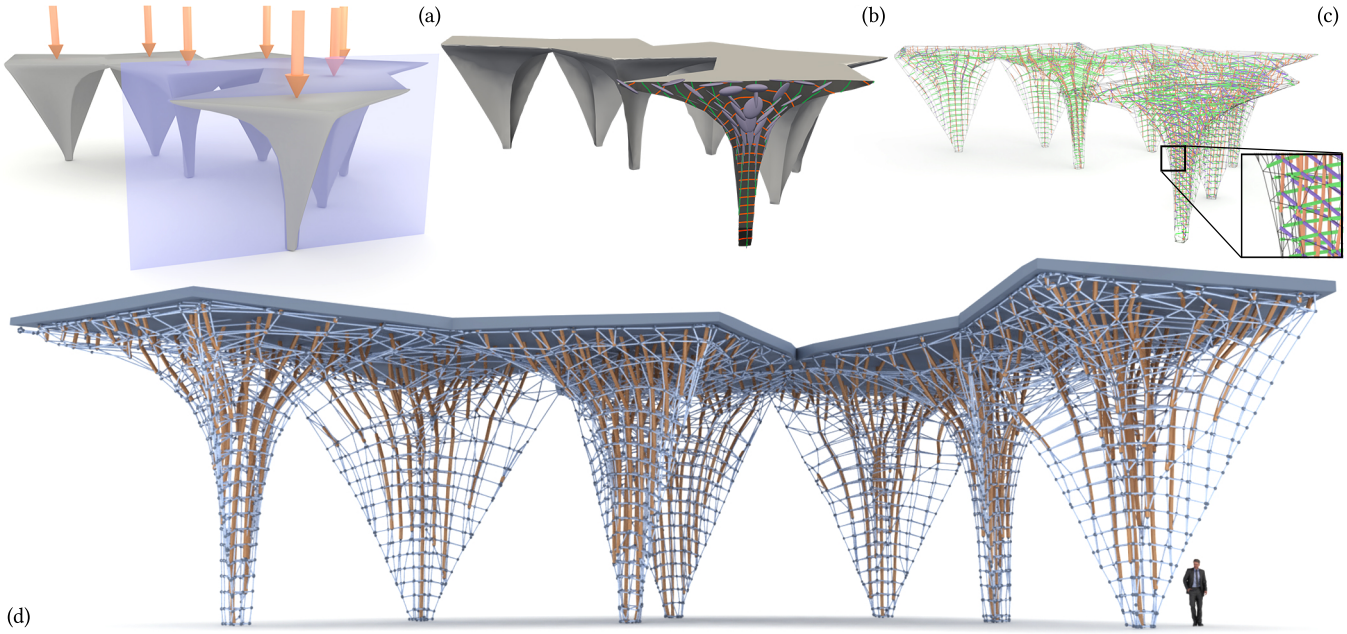


Figure 1: Our method generates design-centric and structurally sound truss structures, with a global parametrization. Given an arbitrary 3D domain (a) with static loads (orange arrows), we compute a globally consistent frame field aligned with the principal stresses. The frames are integrated to form a stress-aligned global parametrization (b) whose isocurves are extracted to form *Michell Trusses* (c). The global parametrization facilitates efficient edits by designers, architects and engineers, such as structurally reinforcing the major compression direction by varying the element thickness and utilized material (d).

SCF '19, June 16–18, 2019, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-6795-0/19/06…$15.00
https://doi.org/10.1145/3328939.3328999

ABSTRACT

We present the first algorithm for designing volumetric Michell Trusses. Our method uses a parametrization-based approach to generate trusses made of structural elements aligned with the primary direction of an object's stress field. Such trusses exhibit high strength-to-weight ratio while also being parametrically editable which can be easily integrated with parametric editing tools such as Autodesk Fusion. We show a number of examples that demonstrate that the output of our algorithm produces truss structures

that are aligned with an object's underlying stress tensor field, are structurally sound and that their global parametrization facilitates the creation of unique structures in a number of domains.

CCS CONCEPTS

• **General and reference → Design**; • **Computing methodologies → Physical simulation**; *Mesh models.*

KEYWORDS

curve networks, design, simulation, topology optimization

ACM Reference Format:
Rahul Arora, Alec Jacobson, Timothy R. Langlois, Yijiang Huang, Caitlin Mueller, Wojciech Matusik, Ariel Shamir, Karan Singh, and David I.W. Levin. 2019. Volumetric Michell Trusses for Parametric Design & Fabrication. In *SCF '19: Symposium on Computational Fabrication (SCF '19), June 16–18, 2019, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3328939.3328999

1 INTRODUCTION

The primary objective of engineering, it is sometimes said, is to develop the stiffest possible structure while using the least amount of material [Doubrovski et al. 2011]. This guiding principle can be seen in many everyday structures such as bridges and stadiums. Strength-to-weight trade-off is naturally expressed as an optimization problem and its solution has become a foundational challenge in mathematics, computer science, and engineering.

Almost all structural optimization algorithms discretize the material distribution within the structure and then attempt to sparsify this distribution (see Figure 4). The nature of this discretization, be it voxels, level-sets, or trusses, gives birth to specific optimization methods. The outputs of current methods (see Figure 2, 3), however, do not integrate automatically into a traditional engineering design process, that is rooted in parametric computer-aided-design (CAD) tools [OYuce et al. 2010]. Parametric design has numerous advantages: parameter values can be varied for intuitive shape and materials edits; parameteric models are easier to render as abstractions that aid design visualization; and parametrizations offer better integration with manufacturing processes, such as optimized tool bits and paths for improved quality and lower build time. The Antonio Gaudi and Frei Otto inspired parametric and generative design trend of *form finding*, combining structural and functional constraints with aesthetic goals [Lachauer et al. 2010], reaffirms the need for a parametric structural optimization solution that integrates with the design process.

Unfortunately, all previous structural optimization algorithms produce outputs that do not easily admit a global parametric representation. Instead, the parametrization is often constructed via time-consuming, manual intervention such as tracing over optimized results (Figure 2). Recent work [Bandara et al. 2016] has attempted to generate CAD-friendly subdivision-surfaces, but any deviation from the input topology still requires considerable manual intervention. Worse still is the lack of editing options other than manipulating raw output at the element level (e.g. voxels, tetrahedra, triangles, see Figure 3). As such, while the notion of topology optimization is a hot topic in both research and mainstream CAD, it is poorly integrated into current industrial practice.

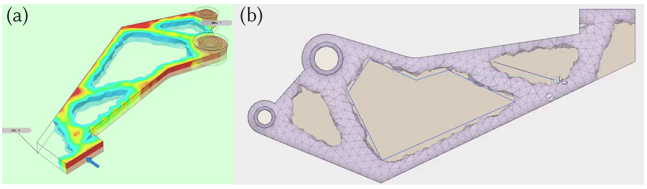


Figure 2: Topology optimization results can be challenging to fabricate. Even when using state of the art commercial tools for topology optimization such as Fusion 360 [Autodesk 2018] (a), users have to manually trace over the results (b) to produce a geometry fabricable via non-additive manufacturing techniques. © Autodesk Sustainability Workshop https://youtu.be/lyTULzvHhXw. Used under CC BY 3.0.

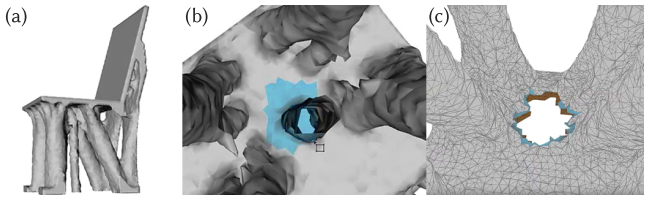


Figure 3: Lacking a parametric representation, topology optimization results can be tiresome to edit. Removing undesirable features (chair's fifth "leg", for example) of a topology optimized geometry (a) can require tedious manual processing such as deletion of individual mesh elements (b) and manual repair of topological defects (c). © Frédéric SINFORT https://youtu.be/0lIqEemsmcU. Used under CC BY 3.0.

Our goal is to bridge the gap between design and structural optimization via an algorithm that produces parameterized output by construction, allowing seamless parametric editing. To do this, we deviate from the standard approach of sparsifying a material distribution directly and instead generate a global parametrization that can be used to organize geometry in a structurally sound way. Michell [1904] laid the mathematical foundations for creating such artifacts by proving that for a given material budget, all elements of the optimal (stiffest) structure must follow paths of maximum strain. Such structures are called *Michell Trusses.*

Our method attempts to enforce Michell's criterion by constructing a global, vector-valued function such that the coordinate lines follow the directions of the virtual principal stresses. Once constructed, we can trace the coordinate lines or planes of this parametrization and construct structural elements such as tubes or sheets along those to produce structurally sound outputs.

Our procedure yields the first algorithm for producing globally parameterized, discrete, Michell Trusses inside arbitrary 3D domains. We show results on various 2D and 3D examples, which demonstrate the high strength-to-weight ratio we achieve compared to naïve truss layouts. Perhaps more importantly, we will shat that trusses generated by our method are not just structurally sound, but they can easily be edited downstream as well. Examples will include efficient global selection and editing operations to aid

Table 1: Compared to existing approaches, our method generates a global parametrization over the input domain, resulting in a truss structure that is easy to scale, amenable to CAD-based post-processing, and simple to understand and manipulate.

Method (*Parametrization*)	Output Topology	Multiscale Result	CAD-Compatibility	Output Complexity
Continuum-based (*none*)	Unconstrained	No	Manual resketching	High-dimensional simplicial mesh
Ground structure (*local*)	Initialization-dependent	No	Import	Edge-weighted graph
Ours (*global*)	Locally fixed	Yes	Import & post-process	3 curve families

design tasks, sparsifying output post-facto for aesthetic and fabrication considerations, allowing users to easily delete elements of the design as well as to add new structural or aesthetic components and finally, allow composition of several optimal trusses into larger objects. Performing such customizations with existing methods is essentially an impossible task (Table 1). By enabling parametric design directly on optimized output, our algorithm provides the missing, yet much desired, seamless connection between structural optimization and parametric design.

2 RELATED WORK

Structural optimization is a classic problem in computational design, fabrication and digital manufacturing. Methods exist to help designers identify the absolute weakest parts of objects [Zhou et al. 2013] or the weakest parts under real world forces [Langlois et al. 2016]. Other methods attempt to reinforce designs to improve their strength [Schumacher et al. 2018; Ulu et al. 2017] or find the most stable orientation for 3D printing [Umetani and Schmidt 2013].

In this paper, we focus on the problem of generating structurally sound objects via optimal material placement. Algorithms for this task define optimality using some measure of an object's strength—most often attempting to minimize an object's compliance under a given load [Bendsøe and Sigmund 2009; Freund 2004] while satisfying constraints on the amount of material used. A large body of works in this domain optimize over a continuum of material, typically discretized as voxels or level sets. Lu et al. [Lu et al. 2014] utilize a volumetric voronoi diagram to carve cavities inside objects and improve strength-to-weight ratios. We refer the interested reader to the excellent survey by Deaton and Grandhi [2014] on continuum-based approaches, and focus on the more-pertinent truss-based optimization methods here.

2.1 Truss Optimization

Truss-based optimization methods [Bendsøe et al. 1994; Freund 2004; Wang et al. 2013] are attractive for their small number of design variables (compared to voxel-based methods) and ease of manufacturing. Michell [1904] first discovered that an optimal truss layout (in terms of strength-to-weight ratio) is given when trusses are aligned with the principal stress directions induced by loading conditions. Intuitively, this aligns elements with the directions of pure compression and tension minimizing stress due to bending. In certain cases, it is possible to solve for this optimal layout analytically [Jacot and Mueller 2017], but no analytical solution is known for the general case, so the ground structure method (GSM) [Dorn et al. 1964; Pedersen 1993; Zegard and Paulino 2015] is used. Here, an initial layout of a finite number of trusses is specified, and the radii and connectivity of the trusses are optimized to minimize the total weight. The traditional GSM formulation suffers from several problems: (1) an initial layout of node positions needs to be specified, which can limit the solution space; (2) it can yield self-intersecting beams; (3) it assumes that the cross-section of each truss member can be set independently, making large-scale manufacturing challenging.

Multiple heuristic methods [Camp and Farshchin 2014; Kaveh et al. 2008; Kawamura et al. 2002; Li et al. 2009] have been proposed to address the third problem by limiting the cross-sections of the trusses to a small set. Mixed-integer programming techniques have been used to achieve global optima for this problem [Achtziger and Stolpe 2007; Rasmussen and Stolpe 2008; Stolpe and Svanberg 2003]. However, these methods were only demonstrated on small models. Jiang et al. [2017] recently demonstrated much larger examples by dividing the mixed-integer problem into three subproblems that were solved iteratively. This works well in practice, but does not guarantee a globally optimal solution. The method also optimizes initial node positions and connectivity to avoid self-intersections, but still requires an oversampled initial mesh, the design of which remains challenging. An interesting alternative approach [Norato et al. 2015] utilizes a continuum-based optimization to solve for an optimized truss. However, it requires the user to provide a pre-determined set of input beams, and only optimizes for their positions and connectivity.

While the methods above try to approximate a truss layout via optimization [Bendsøe et al. 1994], other methods more directly attempt to generate Michell layouts. Tam et al. [Tam et al. 2015; Tam and Mueller 2017] generate principal stress lines directly by integrating the stress field, and develop a novel robotic arm printer capable of printing along these lines directly. However, currently the method only applies to 2.5D structures (i.e., structures that are 3D but only need one layer of trusses, such as shells or membranes), and cannot handle 3D volumetric cases. Similarly, Pellis and Pottamn [2018] describe a method for aligning curvature of (2.5D) sheets to principal stresses.

Li and Chen [2010] begin with a (very simple) user provided beam network which connects the contacts to the points of application of external forces. Then, an iterative method subdivides this structure and better approximates principal stress lines until the desired compliance/strain energy is achieved. While motivational, this method only works in 2D. It is also unclear if the user interaction is amenable to more complicated shapes or load configurations. Li et al. [2017] produce rib like reinforcements aligned with principle stress directions but again, only for 2.5D structures.

2.2 Parametrization-Based Mesh Generation

Our method replaces structural optimization with automatic mesh generation and provides the first algorithm for computing discrete Michell Trusses in arbitrary 3D domains under arbitrary loads.

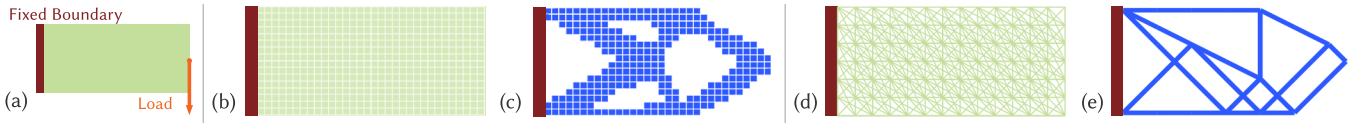


Figure 4: Optimization of a cantilever beam (a) using voxel-based continuum-method (c) and a ground structure method (e). Voxel-based continuum-methods optimize for material placement (c) on the voxelized domain (b), while the ground structure method over samples the domain with truss members (d) and solves for the optimum set of members (e).

Our approach is inspired by recent developments in hex and quad meshing for 3D geometries (e.g., [Nieser et al. 2011; Panozzo et al. 2014]). These algorithms use prescribed frame fields to align the gradients of a volumetric function such that a hex mesh can be extracted. The general hex meshing problem is hard and still an active area of research. None of the currently available algorithms satisfy the criteria necessary for solving our particular problem.

The seminal paper, CubeCover [Nieser et al. 2011], solves a generalized version of the parametrization and mesh extraction problem we solve. However, their method must introduce discrete optimization variables in order to compute a well aligned frame field. We confirmed via personal communication that CubeCover does not generate appropriate frame fields and thus "it's impossible for the software in its current state to be used as stand-alone solution" [Nieser 2018]. Ray et al. [2016] and Solomon et al. [2017] tackle the issue of frame field generation by introducing functional representations of frames. Ray et al. align their frame field with a mesh boundary and smoothly interpolate into the object volume. Solomon et al. also smoothly interpolate inside the object volume using a boundary element based approach. However, our problem requires the opposite objective as we care little for boundary alignment and much more for accurate alignment within the mesh volume itself. L_p-centroidal voronoi tessellation [Lévy and Liu 2010] can generate hex-dominant meshes that take a background anisotropy field into account. However, it does not follow the anisotropy as closely as our method does, and it also does not provide enough information to generate isocurves which can be manipulated. Our method explicitly gives end-to-end isocurves, which can be used for manipulation and sparsification—something which has proven invaluable in generating fabricable results (Fig. 13).

Lyon et al. [2016] present a method for mesh extraction; that is, given a parametrization on a tetrahedral mesh, they extract out a 3D-embedded graph. However, their method requires that the input parametrization is boundary-aligned. Again, our method requires good alignment in the interior of the object, not the boundary, making this method unsuitable. Many of our results such as a bridge or beam (13, 14) are naturally stronger when trusses are not normal to the boundary. Unlike all of the methods above, ours is the first to generate global, structurally sound parametrizations.

3 BACKGROUND AND PRELIMINARIES

We now provide the technical background needed to understand our formulation, beginning with an introduction to truss optimization.

3.1 Truss Optimization

A *truss* is a structure consisting of a network of members, each of which is under purely axial stress. Typically, the forces only act at the joints between these members, known as the *nodes* of a truss. Given a domain $\Omega \subset \mathbb{R}^d$, and boundary conditions consisting of a set of static forces applied on the boundary and anchoring parts of the boundary to fixed supports, truss optimization is the problem of finding a structurally sound truss minimizing the volume of material utilized. In general, this involves optimizing over three sets of design parameters:

(1) connectivity of the truss members (the *topology*);
(2) positions of the nodal points (the *geometry*); and
(3) cross-sectional areas of members (the *size*).

We use *truss layout* to refer to the truss topology and geometry.

In the classical truss optimization formulation, known as the Ground Structure method, an a priori chosen set of uniformly spaced nodal points and members cover the problem domain Ω, forming the so called *ground structure*. The topology of the optimal truss is generated by varying the cross-sectional areas of the members, allowing for zero areas which effectively removes those members. Since the nodal points are assumed to be fixed, the classical approach only solves for the topology and size parameters. In contrast to this, we focus on the first two problems while enabling intuitive user adjustment of truss size.

3.2 Michell Truss Theory

Michell's theorem [1904] characterizes the fundamental properties of the optimal truss structure, called the Michell Truss, for the problem defined above. The theorem states that the members of the optimal truss structure lie along the *principal directions* of the *virtual* stress field. This is the stress field induced by the given external forces if the domain were to be uniformly filled with material, and principal stress directions simply refer to the eigenvectors of the stress tensor matrix. Owing to the continuity of the stress field and the orthogonality of eigenvectors of a Hermitian matrix, the principal stress directions form a set of families of orthogonal curves called the principal stress lines. In the continuous setting, an optimal Michell Truss consists of an infinite set of infinitesimally small members tracing these curves. Computationally, the optimal truss layout for the chosen discretization consists of finite sized members approximating the principal stress lines (Figure 5).

In the Ground Structure method, the approximation error is determined by the density and connectivity of the initially chosen ground structure. Unfortunately, it is difficult to choose an appropriate discretization balancing accuracy and computational time for complex domain geometries.

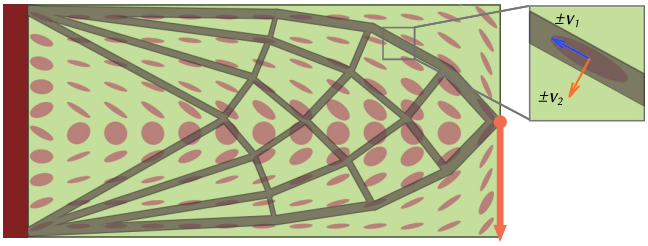


Figure 5: Michell Truss members are aligned with the principal directions of the underlying stress field. Here, we show the example of a cantilever. The ellipses visualize the stress tensor, and the optimal Michell Truss for a chosen discretization is overlaid on the problem domain. Inset: Michell Truss member aligned with a stress eigenvector.

4 STRESS-ALIGNED TRUSS GENERATION

We take a parametrization-based approach to generating stress aligned trusses. Our algorithm (Figure 6) consists of four independent phases: (1) stress field generation using finite element analysis (FEA), (2) stress-aligned smooth frame field fitting, (3) volumetric texture parametrization, and (4) structural member extraction.

One might ask, why follow such an approach, given that robust and reliable hex-meshing for arbitrary geometries is, as yet, unsolved. Fortunately, our problem is more amenable to a solution than that of general hex meshing as we have a volumetric stress tensor field to guide us. Moreover, unlike field-aligned meshing problems, we do not require a boundary aligned structure.

These considerations eliminate some of hex-meshing's most aggravating difficulties, allowing us to develop a flexible algorithm, which, as we will demonstrate, can be applied to a wide variety of geometries. In the remainder of this section, we will detail each step of our truss generation method.

4.1 Finite Element Analysis

The first step of our method is to generate a stress tensor field for an input geometry. We begin with standard linear elastic [Gould and Feng 1994] finite element analysis with tetrahedral discretization [Belytschko et al. 2013; Levin et al. 2017] for this task. Both Dirichlet and Neumann boundary conditions are applied based on the expected loading conditions of the given shape. We then compute the Cauchy stress tensor field (for our elements, a single tensor per tetrahedron) for use in subsequent algorithmic stages. We use the same discretization for all steps of the method.

A natural consequence of using FEA with linear basis functions is that while the resulting displacement field is defined on nodes and is thus piecewise linear, the stress field is piecewise constant with discontinuities at the boundaries of the mesh tetrahedra. The rest of our method treats the stress-field as the ground truth, and as such, stress discontinuities (Figure 7) can degrade the output of the downstream steps. In general, one could use higher-order elements to achieve a continuous stress field. However, our isocurve extraction step (subsection 4.4) requires linear FEA, and therefore we chose to smooth our stress field using Loubignac iterations [Loubignac et al. 1977]. This iterative method converges to a continuous stress field defined at the nodes of the mesh, which we then reproject to the centroids of the mesh elements using barycentric coordinates.

In the remaining sections, we refer to the continuous input domain as $\Omega \subset \mathbb{R}^3$ and the tetrahedral discretization of the domain as the mesh $\mathcal{M} = (\mathcal{V}, \mathcal{T})$.

4.2 Stress Aligned Frame Field Generation

Naïvely, a Cauchy Stress tensor field $\sigma(\mathbf{x})$ can be interpreted as a frame field by representing each tensor by its three eigenvectors. However, this is not true in general since the frame defined by a degenerate tensor is not unique. Even without tensor field degeneracies, a frame field encoded as a rotation matrix is still almost certain to be non-smooth as the direction of each eigenvector can be arbitrarily flipped or interchanged. Previous algorithms for frame-aligned parametrization [Nieser et al. 2011] handle such symmetries by searching over all possible symmetric frame configurations to implicitly compute a continuous frame field. This is effective but complicates optimization by introducing discrete variables.

Therefore, we desire an optimization scheme that a) avoids discrete variables and b) takes tensor field degeneracies into account. Here, we are influenced by methods which work with inherently symmetric, functional representations of frame fields [Ray et al. 2016; Solomon et al. 2017]. Our optimization functional is inspired by the work of Levin et al. [2011] which shows that by using the Rayleigh Quotient [Horn and Johnson 1990] as an objective, one can produce vector fields that smoothly align with the most locally anisotropic direction of a tensor field.

A 3D frame can be encoded using three unit vectors. We define the notion of alignment of a single unit vector with the stress tensor using the square root of the absolute value of the Rayleigh Quotient, and recall the notation $\|\cdot\|_M$, where $M \in \mathbb{R}^{3\times3}$ is a positive symmetric matrix. For unit vectors $\mathbf{v}$, the norm induced by M, $\|\mathbf{v}\|_M = (|\mathbf{v}^T M \mathbf{v}|)^{1/2}$, is maximized when $\mathbf{v}$ is aligned with the primary eigenvector of M.

Let the eigendecomposition of σ be given by $\sigma = Q\widetilde{\Lambda}Q^T$, where $\widetilde{\Lambda}$ is a diagonal matrix whose diagonal elements are the sorted (in decreasing order) eigenvalues of σ. We define

$$\sigma_+ = Q\Lambda Q^T, \tag{1}$$

where the matrix Λ is obtained by linearly rescaling the diagonal values of $\widetilde{\Lambda}$ to a range $[1, k]$ (we use $k = 10$) using

$$\Lambda_{ii} = 1 + (k-1)\frac{\widetilde{\Lambda}_{ii} - \widetilde{\Lambda}_{33}}{\widetilde{\Lambda}_{11} - \widetilde{\Lambda}_{33}}. \tag{2}$$

For the special case when $\widetilde{\Lambda}_{11} = \widetilde{\Lambda}_{33}$ up to machine-precision, we set $\Lambda_{ii} = k\ \forall i$. This rescaling ensures that σ_+ is positive-definite, while its eigenspaces are the same as that of σ.

We are now ready to define alignment of frames and positive-definite tensors. Intuitively, each of the three vectors of a tensor-aligned frame should individually align with an eigenvector of the tensor. Let the eigenspaces of σ_+ associated with unique eigenvalues be $\mathsf{E}_1, \mathsf{E}_2, \ldots$ of dimension $d_1, d_2, \ldots$. Recall that the dimension of the eigenspace is the same as the multiplicity of the corresponding eigenvalue. A frame represented as a rotation matrix $R = [\mathbf{r}_1\mathbf{r}_2\mathbf{r}_3]$ is called an optimally-aligned frame for a positive-definite tensor σ_+, if each eigenspace E_i contributes d_i columns to R. We prove in

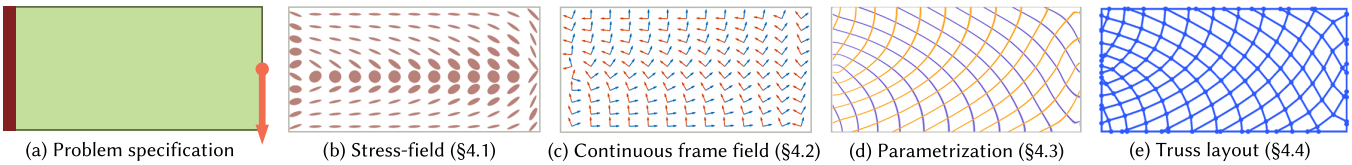


Figure 6: Overview of our method: (a) starting with a problem domain (green) with fixed points (red) and loads (orange) as boundary conditions, we perform FEM analysis to compute the stress field (b). A continuous and smooth frame field (c) aligned with the principal stresses is then computed. The components of this frame field are used to compute a texture parametrization on the domain (d), whose isocurves (shown in orange and purple) are traced to extract a stress-aligned truss structure (e).

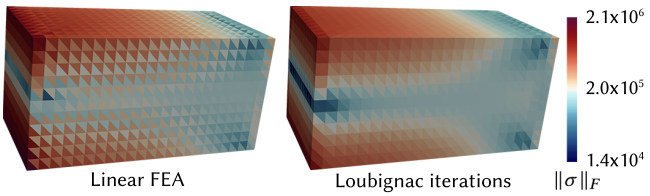


Figure 7: Loubignac iterations [1977] give a continuous stress field (right) by smoothing the discontinuous stress field generated by linear FEA (left).

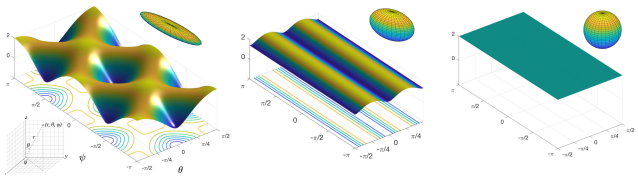


Figure 8: Our cost function has identical local minima corresponding to any orthogonal transformation that aligns a frame with the second and third eigenvectors of a tensor. This behavior is consistent for tensors with 3 (left) 2 (center) and 1 (right) distinct eigenvalues.

the supplemental material that the set of frames aligned with σ_+ is exactly the set of minimizers of the function

$$E\left(R = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)\right) = \sum_{j=1}^{3} \|\mathbf{r}_j\|_{\sigma_+}. \tag{3}$$

However for the Michell Truss problem, we prefer a fit between a frame and a stress tensor as one where the first axis of the frame is aligned with the primary eigenvector of the stress tensor and the other two axes are aligned with the second and third eigenvectors (though it does not matter which aligns with which). This definition of alignment prefers that the frame vector aligned with the primary principal stress direction remains the same, thus resulting in smoother frame-field aligned parametrization (§4.3). To this end, we define a "truncated" frame-tensor matching function:

$$E^i_{data}\left(R = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)\right) = \|\mathbf{r}_2\|_{\sigma^i_+} + \|\mathbf{r}_3\|_{\sigma^i_+}, \tag{4}$$

where $\sigma^i \in \mathbb{R}^{3\times3}$ is the stress tensor of the i^{th} tetrahedron in our mesh $\mathcal{M}$ (we use the simulation discretization for the fitting stage as well) and the positive-definite matrix σ^i_+ is defined according to Equation 1. Note that the choice of the largest positive eigenvalue is arbitrary; one could choose to fix alignment with the smallest eigenvalue as well. The aim is just to achieve smoother frame fields. Figure 8 illustrates that this cost function has a set of identical minima at every frame alignment which satisfies our criteria. In order to ensure good numerical behaviour, we experimentally choose $k = 10$ in Equation 2 across all the tets.

Next we need a method for disambiguating the local minima in Eq. 4. Typically this is done combinatorially, but here we follow the approach of Solomon et al. [2017] and instead use a smoothness energy to produce a well-fitted, consistently aligned frame field. Solomon et al. represent rotations using canonical axis functions and use a standard Laplacian smoothing term. While we borrow their smoothing energy, we cannot use their frame representation as it requires an extra, approximate projection to produce proper, orthogonal frames. In our problem, alignment with the data is critical, and introducing error via such a projection is unacceptable.

Instead we represent a frame using rotation vectors $\omega_j \in \mathbb{R}^3$ stored at each vertex, with the subscript j indexing the vertex. The rotation matrix at the centroid of a tetrahedron $T^i = (v_0, v_1, v_2, v_3)$ is then obtained via the matrix exponential

$$R^i = \text{expm}\left(\left[\frac{\sum_{j=0}^{3} \omega_{v_j}}{4}\right]_\times\right) \in \text{SO}(3),$$

where the $[\cdot]_\times$ operator computes the cross product matrix from an input vector. The rotation vector representation also allows us to define a smoothness energy in the following manner:

$$E_{smooth}\left(\omega\right) = \frac{1}{2}\omega^T L\omega \; + \; \frac{1}{2}\omega^T\omega, \tag{5}$$

where ω is the stacked vector of all per-vertex ω_j, and L is the block diagonal cotangent Laplacian matrix. The second term regularizes ω and prevents it from taking arbitrarily large values. In practice we found this helped with the stability of the line search of our optimization scheme (`fmincon` [The MathWorks, Inc. 2018]).

Note that our energy indicates another point of departure from hex-dominant meshing methods such as CubeCover [Nieser et al. 2011] and Solomon et al. [2017]—our aim is to optimize for a globally consistent frame-field. This allows us to solve for a globally continuous frame-aligned parametrization in the next step. Such a global parametrization is in contrast to the aforementioned methods which utilize local parametrizations demarcated by the singularities of the frame-field, in order to prioritize low cell distortion and boundary alignment. Since we don't care about these objectives and prefer a global parametrization for ease of user control, we "smooth

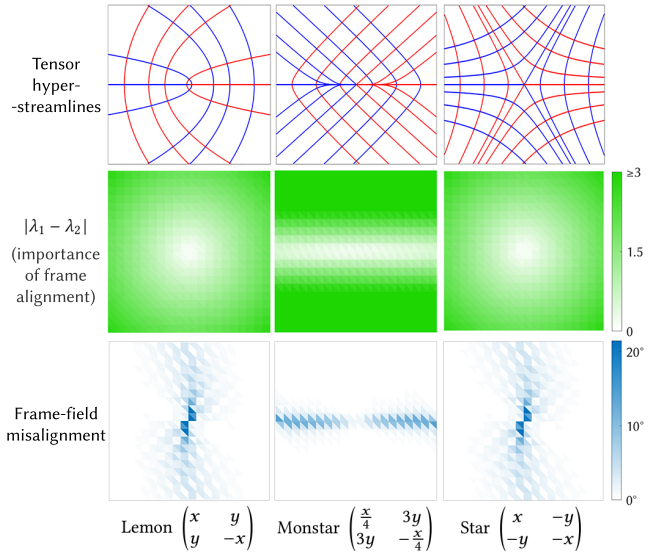


Figure 9: We solve for a globally-consistent frame-field by "smoothing out" tensor field singularities. Here, we show the behaviour of our frame-field solve on the three standard 2D tensor-field singularities (lemon, monstar, and star). Notice that our frame field deviates from the tensor's eigenvectors where $|\lambda_1 - \lambda_2| \approx 0$ and therefore, alignment is not important. Example tensor fields and hyperstreamline visualization (top row) taken from Liu et al. [2008].

out" tensor-field singularities by slightly misaligning with the tensor field near singularities (Figure 9). This is somewhat analogous to hex-only meshing using Polycube Maps [Gregson et al. 2011; Tarini et al. 2004] which also seek to build global parametrizations.

4.2.1 Optimization Details. Initially we attempted to perform frame fitting using a weighted sum of Equation 4 and Equation 5:

$$E_\alpha(\omega) = \sum_{i=1}^{N} E^i_{data}\left(\boldsymbol{r}_1(\omega), \boldsymbol{r}_2(\omega), \boldsymbol{r}_3(\omega)\right) + \alpha E_{smooth}(\omega) \quad (6)$$

$$\omega^* = \underset{\omega}{\operatorname{argmin}}\ E_\alpha(\omega) \quad (7)$$

where $N = |\mathcal{T}|$ is the number of tetrahedra in $\mathcal{M}$, α is a scalar weight, and $R^i = (\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3)$. Minimizing this cost function, using an L-BFGS Hessian approximation, revealed issues in choosing an appropriate α. To alleviate this problem, we again lean on the fact that our sole concern is minimizing the data term. The only purpose of the smoothness energy is to help us choose an appropriate local minima to descend into. To this end, our final fitting algorithm is Augmented Lagrangian-esque [Nocedal and Wright 2006] in that we repeatedly minimize Equation 7 with increasingly smaller α until the data cost stops decreasing. In practice our termination criteria was not complex: a fixed thirty iterations after each of which α was reduced to $(2/3)\alpha$. This proved to be more than enough for all our examples and yielded excellent results. There is a minor implementation detail which arises when using matrix exponentials as a parametrization of rotation matrices—their gradient is undefined at $\omega = \mathbf{0}$. We avoid this problem by perturbing any 0 length angular velocity vector by the square root of machine epsilon (a standard work-around). Using the Taylor series expansion around zero (see [Grassia 1998]) could provide a more robust fix for this problem. However in practice, our simple fix did not cause any issues. We believe this is because other terms in the gradient ensure good numerical behavior near the singularity.

4.3 Global Parametrization Computation

We use our smooth, data-aligned frame field to compute a stress-aligned parametrization from which we will create a Michell Truss. We define $\mathcal{U} \subset \mathbb{R}^3$ as a volumetric texture domain. Recall that $\Omega \subset \mathbb{R}^3$ is the world space that our object occupies. We chose our structural members to lie along the coordinate lines of $\mathcal{U}$ and seek to find a continuous parametrization $\phi(\mathbf{x}) : \Omega \to \mathcal{U}$ that aligns these coordinate lines with our frame field $R : \Omega \to \mathrm{SO}(3)$. Formally we seek a $\phi(\mathbf{x})$ such that

$$\begin{aligned} \frac{\partial \phi}{\partial \mathbf{x}} \mathbf{r}_1(\mathbf{x}) &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \\ \frac{\partial \phi}{\partial \mathbf{x}} \mathbf{r}_2(\mathbf{x}) &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \\ \frac{\partial \phi}{\partial \mathbf{x}} \mathbf{r}_3(\mathbf{x}) &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \end{aligned} \quad (8)$$

everywhere on Ω, where $\mathbf{r}_i(\mathbf{x})$ gives the i^{th} column of $R(\mathbf{x})$.

For the discrete setting, we can write these requirements as a linear system of equations by constructing the discrete directional gradient operator for each tetrahedron in our mesh:

$$G^i(\mathbf{v}) = \left[\mathbf{v}^i_x \cdot G^i_x + \mathbf{v}^i_y \cdot G^i_y + \mathbf{v}^i_z \cdot G^i_z\right], \quad (9)$$

where G_x, G_y and G_z are the discrete gradient operators of our tetrahedral mesh, $\mathbf{v} \in \mathbb{R}^3$ is the direction in which the derivative is to be measured (at the centroid of a tetrahedron) and i indexes our tetrahedra. We can assemble these local directional derivative operators into global matrices to produce the global operator $G(\mathbf{v})$.

We proceed by constructing three directional derivative operators, one for each frame director

$$\begin{aligned} \mathbf{G}_1 &= G(\mathbf{r}_1) \\ \mathbf{G}_2 &= G(\mathbf{r}_2) \\ \mathbf{G}_3 &= G(\mathbf{r}_3) \quad . \end{aligned} \quad (10)$$

Then, we note that Equation 8 can be interpreted as two independent objectives—requiring the gradients of the parametrization to follow the frame field and a regular, equi-spaced, solution. Therefore, in the discrete setting, we formulate the problem as an unconstrained weighted quadratic minimization.

$$\begin{aligned} \phi^* = \underset{\phi}{\operatorname{argmin}}\ \beta \left\| \begin{bmatrix} \mathbf{G}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{G}_3 \end{bmatrix} \phi - \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix} \right\|^2_2 \dots \\ + \left\| \begin{bmatrix} \mathbf{0} & \mathbf{G}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{G}_1 \\ \mathbf{G}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{G}_2 \\ \mathbf{G}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_3 & \mathbf{0} \end{bmatrix} \phi - \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \right\|^2_2 \end{aligned} \quad (11)$$

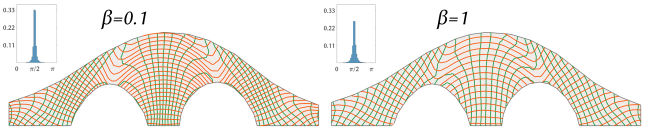


Figure 10: Planar slices of arch bridge parametrizations using different values of β. A low value (0.1, left) prefers orthogonality of parameter lines over equal spacing, while a higher value ($\beta = 1$, right) sacrifices orthogonality for more equally spaced parameter lines. We statistically confirm this effect by plotting the inter-element angle histogram for the trusses obtained by tracing these parameter lines.

where the parameter β provides user control over the regularity of the truss spacing. Figure 10 shows how β influences the solution.

4.4 Truss Layout Extraction

In the final step of our algorithm, the parametrization is used to extract the truss layout as a graph embedded in Ω. In order to avoid confusion with the vertices and edges of the input geometry, we exclusively use the words *nodes* and *elements* to refer to the vertices and edges of the graph extracted from the parametrization. Similar to parametrization based approaches for hex-meshing [Lyon et al. 2016], our aim is to trace the integer isocurves of the parametrization. That is, we want the nodes $\mathcal{N}$ of the extracted graph to be the points mapped to integers, i.e.,

$$\mathcal{N} = \{\mathbf{x} \in \Omega \mid \phi(\mathbf{x}) \in \mathbb{Z}^3\}, \tag{12}$$

and the elements $\mathcal{E}$ connect adjacent points on an integer grid, i.e.,

$$\mathcal{E} = \{\{\mathbf{x}, \mathbf{y}\} \mid \mathbf{x}, \mathbf{y} \in \mathcal{N},\ \phi(\mathbf{x}) - \phi(\mathbf{y}) \in \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}\}, \tag{13}$$

where $\mathbf{e}_i$'s are the standard basis vectors.

Note that only the gradient directions of our parametrization have any physical meaning, and therefore, applying an arbitrary translation and/or scale to the parametrization essentially keeps the physical information intact. In order to provide user control over the density of the extracted truss, we first translate and scale ϕ to normalize it to the range $[0, 1]$, and then scale by a user-defined "resolution" parameter ρ. We refer to this translated and scaled parametrization as $\widetilde{\phi} = (\widetilde{\phi_1}, \widetilde{\phi_2}, \widetilde{\phi_3})$. The detailed steps for tracing the integer isolines of $\widetilde{\phi}$ are described in the supplemental material.

4.5 Implementation Details

Our pipeline is implemented mostly in MATLAB, with the exception of FEA [Levin et al. 2017] which is implemented in C++. Our frame fitting, parametrization, and extraction algorithms are implemented entirely in MATLAB—using `fmincon` to solve the frame fitting optimization and `quadprog` for the texture parametrization. All our code, as well as data, is available at https://github.com/rarora7777/VolumetricTruss.

5 RESULTS

We utilized our method to create globally-parametrized truss layouts for a variety of problem domains. Based on the application, these truss layouts can be utilized by end-users in a number of ways.

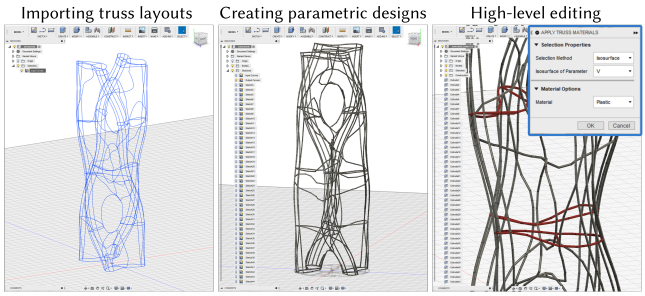


Figure 11: Our Michell Trusses can be easily imported into and edited using a parametric CAD package (Fusion 360 [Autodesk 2018]). The parametric isocurves can either be loaded as piecewise linear truss elements (not shown) or smooth spline curves (left). Users can choose the shape and size of the profile which is extruded along the curves to create Fusion's native "BRep" bodies (center). Beside smooth splines, the global parametrization enables high-level operations such as modifying the material along a certain parameter isosurface (right). UI panel enlarged for clarity (inset).

For example, we wrote plugins for Autodesk Fusion 360 [2018], allowing users to load truss layouts and perform parametric edits directly in Fusion (Fig. 11). Alternatively, users can utilize lightweight scripts to set the parameters. Each element is then replaced with a prism created by extruding a user-defined profile along the element, while nodes are inflated to spheres or convex hulls.

We also show a collection of additional results created using our method and the operations above (Figure 12). This serves to further reinforce the broad applicability of our approach to a wide-range of geometries taken from diverse application domains, such as computer graphics, architecture and aerospace engineering. Our algorithm serves as a drop-in tool which can produce parametric truss structures, suitable for engineering design, easily and robustly.

5.1 Quality of Output Trusses

Michell's Criterion. In order to measure the degree to which our trusses satisfy Michell's criterion, we first compute the misalignment of our frame fields with the stress eigenvectors. For this task, we compute the rotation that aligns the frame with the stress eigenvectors. Given two rotation matrices Q and R, the unique rotation matrix aligning them is given by $S = Q^T R$. However, due to octahedral symmetry inherent in rotation matrices representing frames, we need to compute the *smallest* of 24 possible rotations aligning a frame R to an eigenvector matrix Q (§4.2). By the *smallest rotation*, we refer to the rotation matrix which gives the smallest angle of rotation when converted to axis-angle representation.

Let $\{O_k\}$ ($k \in \{1, \ldots, 24\}$) be the set of matrices in the octahedral symmetry group. The minimum rotation angle between frames represented by matrices Q and R is then given by

$$\theta^* = \underset{k}{\text{argmin}}\ \theta(S_k)$$

where $S_k = Q^T(O_k R)$, and $\theta(S) = cos^{-1}\left((tr(S) - 1)/2\right)$ gives the angle of rotation encoded by an SO(3) matrix S.

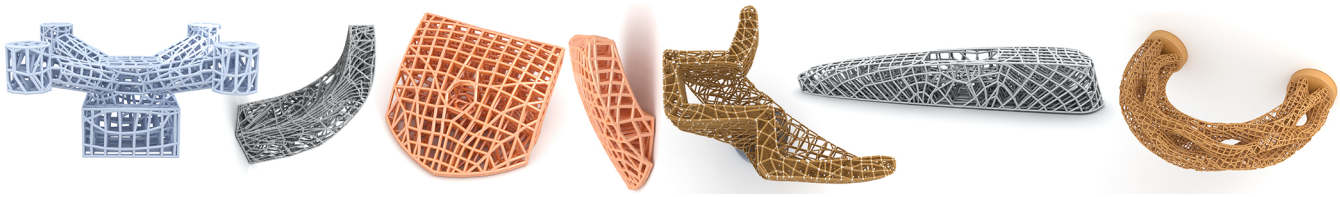
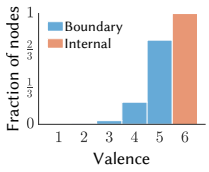


Figure 12: Our method can be utilized for creating optimal structures for a variety of applications. Shown here from left to right—quadcopter frame (aerospace), satellite antenna arm (communications), climbing hold (sports and adventure), bookcase (furniture), helicopter top pylon (aerospace), and *holey* sculpture (fine arts).

We observed excellent alignment between stress tensors and frame fields for our testcases. Table 2 reports the aggregate alignment statistics for each of our problems, while Figure 1b visually depicts the frame-to-stress alignment for the pavilion problem.

5.2 Topology and Geometry of Truss Layouts



Since our parametrization is built from a singularity-free frame field, we expect all the internal (non-boundary) nodes to be *regular*, that is, to have a degree of exactly 6. While our method theoretically guarantees this property, we also **quantitatively verified that the internal nodes for all our test cases were always regular (inset)**. Note that unlike polycube mapping, some of our boundary nodes ($\approx$ 24%) have valence below 5 since we trace sharp edges of the input mesh, as well as the intersections of parameter isolines with such edges. Please see the supplemental material for a detailed description of our isoline tracing procedure.

Next, we test the geometric quality of our truss layouts by measuring the inter-curve angles at each internal vertex. Concretely, at each internal node, we measure the angle between all pairs of incident elements belonging to different parameter isolines. Figure 10 shows that decreasing the parameter β results in closer to orthogonal parametrizations. Table 2 shows the summary statistics for the typical values of β we utilized (1 and 0.1) and shows that our objective of an orthogonal parametrization is well-approximated even with a small value of β.

Table 2: Stress alignment and mesh quality statistics (mean $\pm$ std. deviation) for our truss layouts. All angles are in degrees.

Problem	Frame	Inter-Element Angle	
	Misalignment	$\beta = 0.1$	$\beta = 1.0$
Curved bridge	2.8 ± 2.2	90.1 ± 11.2	90.2 ± 14.9
Mars lander (upper leg)	2.7 ± 2.1	90.3 ± 14.8	90.2 ± 13.1
Satellite antenna arm	2.1 ± 1.9	90.2 ± 16.3	90.1 ± 16.9
Holey pillar	2.6 ± 2.1	90.1 ± 10.2	90.1 ± 11.1
Cantilever beam	1.4 ± 1.3	90.0 ± 5.0	90 ± 8.1
Bar under torsion	1.8 ± 1.8	90.0 ± 13.5	90 ± 11.4
Bar under tension	0.4 ± 0.7	90.0 ± 1.0	90 ± 1.3
Simple bridge	2.2 ± 1.9	90.1 ± 7.2	90 ± 8.4
Mars lander (lower leg)	2.1 ± 1.9	90.1 ± 11	90.3 ± 11.9
Pavilion	2.7 ± 2.1	90.1 ± 10.1	90.1 ± 11.8
Bookcase	2.6 ± 2.1	90.2 ± 17.4	90.2 ± 17.2
Mars lander (body)	2.7 ± 2.1	90.1 ± 13.2	90.1 ± 15.4
Arched bridge	2.1 ± 1.9	90.1 ± 8.8	90.1 ± 12.1
Quadcopter frame	2.5 ± 2.1	90.2 ± 12.8	90.3 ± 15.6
Helicopter top pylon	2.5 ± 2	90.0 ± 9.4	90.1 ± 10.8
Chair (sitting load)	1.9 ± 1.7	90.0 ± 5.4	90 ± 6.2
Chair (rocking load)	2.3 ± 2	90.1 ± 12.9	90.1 ± 14.6
Climbing hold	1.9 ± 1.8	90.0 ± 8.1	90 ± 9
Holey sculpture	2.8 ± 2.2	90.1 ± 9.4	90.1 ± 10.5
Jet engine bracket	2.8 ± 2.2	90.1 ± 11.5	90.1 ± 11.9

5.3 Fabrication and Structural Testing

In order to compute the strength of our trusses, we first union all the individual cylinders and nodal geometries using Libigl's [Jacobson et al. 2016] robust boolean operations to create a watertight mesh. Then we tetrahedralize the mesh using TetWild [Hu et al. 2018] and perform finite element analysis. Fig. 9 in the supplemental shows FEA results on the bridge model and compares the strength of our stress-aligned truss with a) a truss produced by tracing out isolines of the trivial grid-aligned parametrization, and b) boundary-aligned truss created using hex-meshing [Gao et al. 2017] without taking the stress field into account. The result clearly shows that our result outperforms both the trivial truss and the non-stress aligned truss.

We also performed mechanical tests to experimentally support our claim that stress-aligned trusses are strong. All examples were fabricated using our in-house Stratasys F170 FDM printer with dissolvable support. We stress that **our examples could be fabricated via other means (Figure 13d)**, however this is the only fabrication device available in our laboratory.

Our initial test case is a cantilever beam composed of 20 cc of material. We compared our optimized beam against both an unoptimized regular truss—a regular grid with cross-bracing on the faces—and a GSM optimized beam [Zegard and Paulino 2015]. In this test our beam failed at 403 N vs. 299 N for the unoptimized beam and 269 N for the GSM beam (Figure 14). The GSM beam's surprising failure was caused by overfitting to the load case, causing it to remove a substantial amount of cross bracing.

We also 3D printed an ABS plastic bridge with 4mm thick members. The structure was optimized for compression from the top. Weighing 140 grams, it was able to withstand the weight of an adult human weighing approximately 93 kg (205 lbs). Please see Fig. 13b and the accompanying video for the test procedure and results.

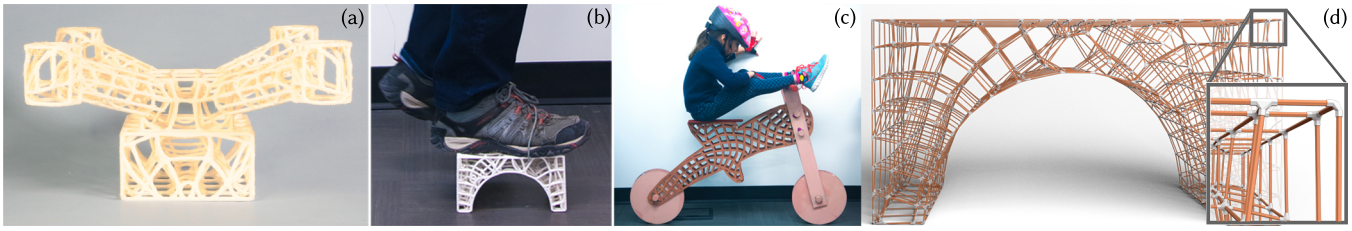


Figure 13: We fabricated our trusses using 3D printing (a, b), and laser cutting (c). The 140g ABS plastic bridge (b) is able to support an adult human weighing 93kg. The laser-cut 2D bike frame (c) was tested with a 5-yr old weighing 21kg. Our trusses can also be assembled using dowel rods and 3D printed joints; shown here as a rendering (d).

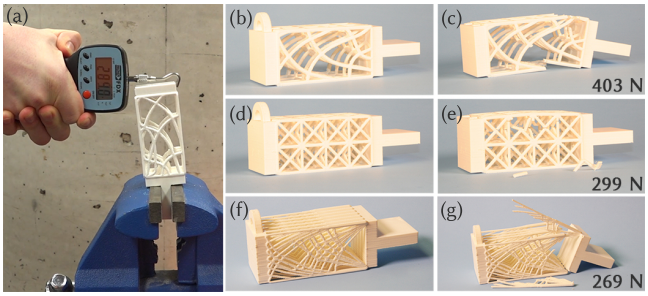


Figure 14: We mechanically tested our optimized bending truss structures (b, j) against a regular grid structure with cross bracing on the faces (d) and a Ground Structure optimized truss (f). We observed failure at 403 Newtons (ours, c), 299 Newtons (unoptimized, e) and 269 N (Ground Structure, g). (a) shows a still from the test in-progress; see the accompanying video for the full test.

We also utilized laser cutting to build an optimized frame for a child's wooden bike (Fig. 13c) using 1/4in. thick Baltic Birch wood. The bike was tested with a 5-yr old weighing 21kg (46 lbs) and no failures occurred. Please see the accompanying video for the test.

Finally, we performed FEM analysis to visualize the stress fields in several of our examples (Figure 15).

6 PARAMETRIC POST-PROCESSING

Unlike existing approaches for structural optimization, the labelled end-to-end curves produced by our method make our results amenable to user editing. We describe four editing operations below; and two additional ones in the supplemental material.

6.1 Radius and Density Control

Our global parametric representation makes it simple to allow user editing of both the radius and density of curves in an optimized truss. Figure 16 illustrates the ability to change global curve radius instantaneously, post optimization. Our method also enables fast curve density control which allows a user to easily control the sparsity of an optimized truss as a post process. Typically, we use a high value of the resolution parameter ρ (Section 4.4) to extract a dense truss layout consisting of smooth curves. However, our global curve parameterization allows us to select subsets of these curves to create a final geometry (something that is not possible with previous approaches). This gives designers and engineers control over both the aesthetic of a truss structure and the total number of curves it contains (Figure 17).

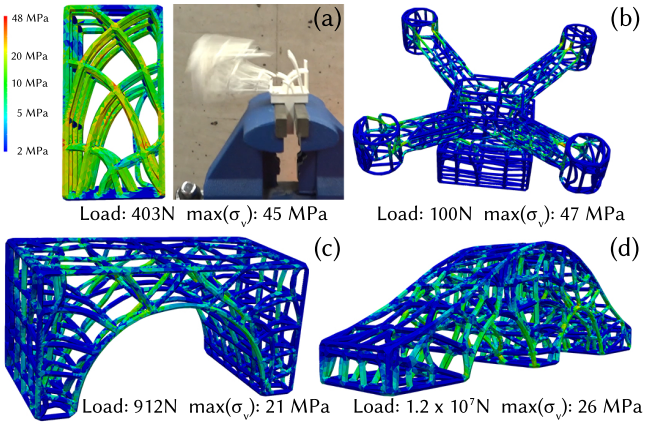


Figure 15: We used linear FEM to simulate our trusses. The material assumed for the visualizations is similar to ABS-M30 plastic: E = 48 MPa, $\nu = 0.45$ (Gauss default), which we used for fabricating the bridge, quadcopter frame, and bending bar. The bending bar yielded at 403N. The simulation (a) shows excellent accuracy in predicting the stress concentrations, which agree with experimentally observed fracture regions. The quadcopter frame is predicted to hold up 100N (10.2kg) of load successfully (b). The miniature flat bridge simulation (c) predicts no fracture at 912N (93kg) load, as confirmed by our experiment. A real-world scale arch-bridge made of 2.5cm thick elements (d) is predicted to hold 1.2×10^7N (10 firetrucks).

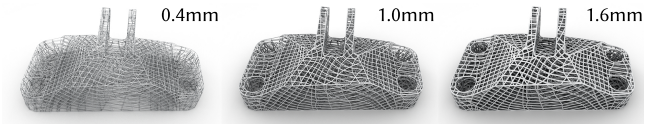


Figure 16: A basic parameter exposed by our method is element thickness; we show three variants of an optimized jet engine bracket here. Testcase motivated by https://grabcad.com/challenges/ge-jet-engine-bracket-challenge.

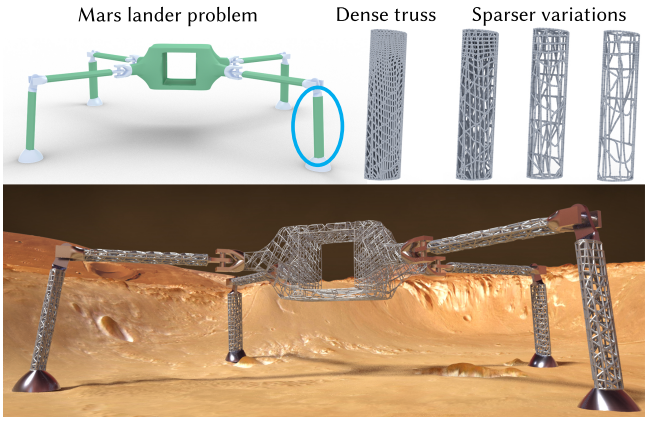


Figure 17: Our stress-aligned trusses naturally encode multi-scale structural information. Truss density can be chosen post-optimization by simply selecting a subset of the extracted parameter isolines. Top row: example truss density options for the mars lander's lower leg. Only the densest (leftmost) structure was obtained via the truss extraction algorithm. Bottom: the finished lander after selecting the parameters for all the optimized parts. Input mesh inspired by NASA's generative design project (https://bit.ly/2PtvzVo).

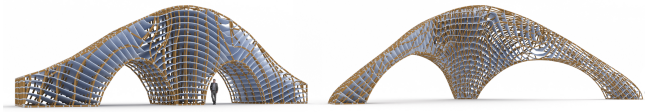


Figure 18: Our global parametrization can enable the creation of stress-aligned sheet structures as well (currently extracted using Paraview's Contour filter [Kitware Inc. 2019]). Here, we show examples of heterogeneous structures—bridges constructed with wooden truss elements combined with sheet metal work across one of the parameter isosurfaces.

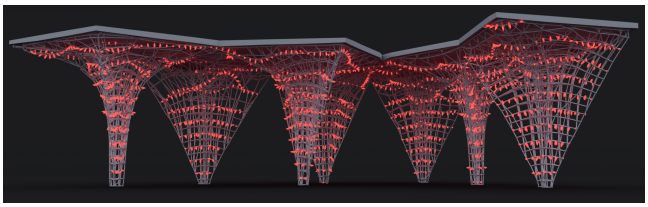


Figure 19: Design utilization of global parametrizations—a designer utilized the consistently labelled curve families to visualize the installation of lights on the pavilion.

6.2 Geometry Modification

Our parameterized structures admit a number of geometry modification operations that are useful from a design perspective. For instance, we can extrude additional geometry or functional elements along parameterized trusses. Figure 19 shows the addition of lights to a pavilion by parallel transporting geometry along a

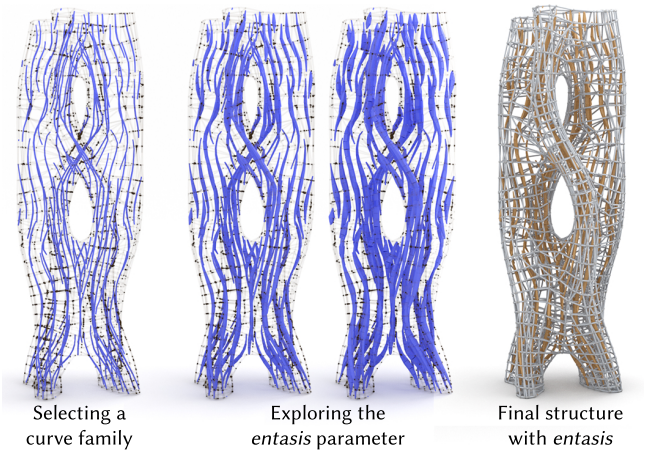


Figure 20: The global parametrization enables a unique combination of structurally optimized forms and aesthetic motifs. For example, an architect selects a curve family (left) to follow the Greek architectural principle of *entasis*—varying the cross-section of structural members to give them a convex-shaped silhouette (right). The workflow involves exploration along a parameter defining the extent to which curves should be thickened in the middle (center).

set of trusses. This operation would painstakingly difficult for the output of typical structural optimization algorithms. We can also add architectural highlights such as *entasis* (Figure 20) in the same manner. Additionally, our method is not limited to extracting field-aligned trusses but can also potentially create field-aligned sheets (Figure 18) which can both improve the structural properties [Sigmund et al. 2016] of a design and provide a unique aesthetic.

7 CONCLUSION

By adopting a parametrization-based approach we have crafted the first algorithm for the design of volumetric Michell Trusses and shown that the algorithm can produce complex, aesthetically pleasing output that is also structurally sound. Our method provides a previously unavailable combination of structural optimality and user control. As such, we believe it serves as an important companion to traditional approaches while also providing engineers, architects, and designers with an exciting new algorithmic tool.

Limitations and Future Work. The most significant limitation of our approach is that it does not incorporate fabrication constraints. Incorporating such constraints into design optimization is an ongoing area of research. We are motivated by recent works to investigate this further [Allaire et al. 2016; Martínez et al. 2018]. We also leave sizing optimization of our trusses as future work, focusing instead on enabling intuitive user selection of truss sizes.

Our method produces trusses whose members are almost equally spaced. This can be problematic if the aim is to preserve surface features in very thin regions of the input geometry.

In truth, we believe that we have only scratched the surface of the user control possibilities of our method. While density specification, vertex snapping, curve removal and geometry generation

are important interactions (and alone facilitate the creation of novel designs), more work is needed to build a powerful user interface for controlling our results. For instance, allowing the user to freely manipulate the truss while providing interactive feedback on the structural soundness of the modified structure is a useful problem to tackle in the future. We would also like to explore incorporating sizing optimization into such an interface [Pérez et al. 2015]. Another potential direction is a more theoretical exploration of how working in finitely-sized domains impacts the stress-line tracing criterion. While our experimental evidence supports that such structures are strong, Michell's original formulation was only given for infinitely-sized domains, where truss members can be positioned anywhere in space.

In general, efficient and fine-grained user control of topology optimization remains an interesting direction for future research. In order to bootstrap further research in this area we are releasing all of our code and data as open-source.

ACKNOWLEDGMENTS

We thank Lawson Fulton and Sarah Kushner for their immense help with rendering the results, Peter Hamilton for narrating the video, and other members of the DGP lab for helping with the structural tests. This research was funded in part by NSERC Discovery (RGPIN-2017-05524, RGPIN2017–05235, RGPAS–2017–507938), NSERC Accelerator (RGPAS-2017-507909), New Frontiers in Research Fund (NFRFE–201), UofT Connaught Fund 03114, Canadian Foundation for Innovations John Evans Leadership Fund, the Ontario Early Research Award program, the Canada Research Chairs program, the Fields Centre for Quantitative Analysis and Modelling, the Adobe Research Fellowship program, and gifts by Adobe Systems, Autodesk and MESH Inc.

REFERENCES

Wolfgang Achtziger and Mathias Stolpe. 2007. Truss topology optimization with discrete design variables—guaranteed global optimality and benchmark examples. *Structural and Multidisciplinary Optimization* 34, 1 (2007), 1–20.

Grégoire Allaire, François Jouve, and Georgios Michailidis. 2016. *Molding Direction Constraints in Structural Optimization via a Level-Set Method.* Springer International Publishing, Cham, 1–39. https://doi.org/10.1007/978-3-319-45680-5_1

Autodesk. 2018. Fusion 360. https://www.autodesk.com/products/fusion-360/overview

Kosala Bandara, Thomas Rüberg, and Fehmi Cirak. 2016. Shape Optimisation with Multiresolution Subdivision Surfaces and Immersed Finite Elements. *Computer Methods in Applied Mechanics and Engineering* 300 (2016), 510 – 539. https://doi.org/10.1016/j.cma.2015.11.015

Ted Belytschko, Wing Kam Liu, Brian Moran, and Khalil Elkhodary. 2013. *Nonlinear finite elements for continua and structures.* John Wiley & Sons.

M. P. Bendsøe, A. Ben-Tal, and J. Zowe. 1994. Optimization methods for truss geometry and topology design. *Structural optimization* 7, 3 (01 Apr 1994), 141–159. https://doi.org/10.1007/BF01742459

Martin P Bendsøe and Ole Sigmund. 2009. *Topology Optimization.* World Scientific.

C.V. Camp and M. Farshchin. 2014. Design of space trusses using modified teaching–learning based optimization. *Engineering Structures* 62-63, Supplement C (2014), 87 – 97. https://doi.org/10.1016/j.engstruct.2014.01.020

Joshua D. Deaton and Ramana V. Grandhi. 2014. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization* 49, 1 (01 Jan 2014), 1–38. https://doi.org/10.1007/s00158-013-0956-z

W. S. Dorn, R. E. Gomory, and H. J. Greenberg. 1964. Automatic design of optimal structures. *Journal de Mecanique* 3 (1964), 25–52.

Z. Doubrovski, J.C. Verlinden, and J.M.P. Geraedts. 2011. Optimal Design for Additive Manufacturing: Opportunities and Challenges. In *Proc. 23rd International Conference on Design Theory and Methodology.* 635–646.

Robert M Freund. 2004. Truss design and convex optimization. *Massachusetts Institute of Technology* (2004).

Xifeng Gao, Wenzel Jakob, Marco Tarini, and Daniele Panozzo. 2017. Robust Hex-dominant Mesh Generation Using Field-guided Polyhedral Agglomeration. *ACM Trans. Graph.* 36, 4, Article 114 (July 2017), 13 pages. https://doi.org/10.1145/3072959.3073676

Phillip L Gould and Yuan Feng. 1994. *Introduction to linear elasticity.* Springer.

F. Sebastin Grassia. 1998. Practical Parameterization of Rotations Using the Exponential Map. *J. Graph. Tools* 3, 3 (March 1998), 29–48. https://doi.org/10.1080/10867651.1998.10487493

James Gregson, Alla Sheffer, and Eugene Zhang. 2011. All-Hex Mesh Generation via Volumetric PolyCube Deformation. *Computer Graphics Forum* 30, 5 (2011), 1407–1416. https://doi.org/10.1111/j.1467-8659.2011.02015.x

Roger A Horn and Charles R Johnson. 1990. *Matrix analysis.* Cambridge university press.

Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. 2018. Tetrahedral Meshing in the Wild. *ACM Trans. Graph.* 37, 4, Article 60 (July 2018), 14 pages. https://doi.org/10.1145/3197517.3201353

Alec Jacobson, Daniele Panozzo, et al. 2016. libigl: A simple C++ geometry processing library. http://libigl.github.io/libigl/.

Benjamin P Jacot and Caitlin T Mueller. 2017. A strain tensor method for three-dimensional Michell structures. *Structural and Multidisciplinary Optimization* 55, 5 (2017), 1819–1829.

Caigui Jiang, Chengcheng Tang, Hans-Peter Seidel, and Peter Wonka. 2017. Design and Volume Optimization of Space Structures. *ACM Trans. Graph.* 36, 4, Article 159 (July 2017), 14 pages. https://doi.org/10.1145/3072959.3073619

A Kaveh, B Farhmand Azar, and S Talatahari. 2008. Ant colony optimization for design of space trusses. *International Journal of Space Structures* 23, 3 (2008), 167–181.

H Kawamura, H Ohmori, and N Kito. 2002. Truss topology optimization by a modified genetic algorithm. *Structural and Multidisciplinary Optimization* 23, 6 (2002).

Kitware Inc. 2019. paraview.simple.Contour — ParaView Documentation. https://kitware.github.io/paraview-docs/latest/python/paraview.simple.Contour.html.

Lorenz Lachauer, Matthias Rippmann, and Philippe Block. 2010. Form Finding to Fabrication : A digital design process for masonry vaults.

Timothy Langlois, Ariel Shamir, Daniel Dror, Wojciech Matusik, and David I. W. Levin. 2016. Stochastic Structural Analysis for Context-aware Design and Fabrication. *ACM Trans. Graph.* 35, 6, Article 226 (Nov. 2016), 13 pages. https://doi.org/10.1145/2980179.2982436

David I.W. Levin et al. 2017. GAUSS: Gaggle of Algorithm for Simulating Stuff. *https://github.com/dilevin/GAUSS* (2017).

David I.W. Levin, Benjamin Gilles, Burkhard Mädler, and Dinesh K. Pai. 2011. Extracting skeletal muscle fiber fields from noisy diffusion tensor data. *Medical Image Analysis* 15, 3 (2011), 340 – 353. https://doi.org/10.1016/j.media.2011.01.005

Bruno Lévy and Yang Liu. 2010. Lp Centroidal Voronoi Tessellation and Its Applications. *ACM Trans. Graph.* 29, 4, Article 119 (July 2010), 11 pages. https://doi.org/10.1145/1778765.1778856

L. J. Li, Z. B. Huang, and F. Liu. 2009. A Heuristic Particle Swarm Optimization Method for Truss Structures with Discrete Variables. *Comput. Struct.* 87, 7-8 (April 2009), 435–443. https://doi.org/10.1016/j.compstruc.2009.01.004

Yongqiang Li and Yong Chen. 2010. Beam structure optimization for additive manufacturing based on principal stress lines. In *Solid Freeform Fabrication Proceedings.*

J. Liu, W. T. Hewitt, W. R. B. Lionheart, J. Montaldi, and M. Turner. 2008. A Lemon is not a Monstar: Visualization of Singularities of Symmetric Second Rank Tensor Fields in the Plane. In *Theory and Practice of Computer Graphics.* The Eurographics Association. https://doi.org/10.2312/LocalChapterEvents/TPCG/TPCG08/099-106

Gilles Loubignac, Gilles Cantin, and Gilbert Touzot. 1977. Continuous stress fields in finite element analysis. *AIAA Journal* 15, 11 (Nov. 1977), 1645–1647. https://doi.org/10.2514/3.7464

Lin Lu, Andrei Sharf, Haisen Zhao, Yuan Wei, Qingnan Fan, Xuelin Chen, Yann Savoye, Changhe Tu, Daniel Cohen-Or, and Baoquan Chen. 2014. Build-to-last: Strength to Weight 3D Printed Objects. *ACM Trans. Graph.* 33, 4, Article 97 (July 2014), 10 pages. https://doi.org/10.1145/2601097.2601168

Max Lyon, David Bommes, and Leif Kobbelt. 2016. HexEx: Robust Hexahedral Mesh Extraction. *ACM Trans. Graph.* 35, 4, Article 123 (July 2016), 11 pages. https://doi.org/10.1145/2897824.2925976

Jonàs Martínez, Samuel Hornus, Haichuan Song, and Sylvain Lefebvre. 2018. Polyhedral Voronoi diagrams for additive manufacturing. *ACM Transactions on Graphics* 37, 4 (Aug. 2018), 15. https://doi.org/10.1145/3197517.3201343

A. G. M. Michell. 1904. LVIII. The limits of economy of material in frame-structures. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 8, 47 (1904), 589–597. https://doi.org/10.1080/14786440409463229 arXiv:https://doi.org/10.1080/14786440409463229

Matthias Nieser. 2018. personal communication.

M. Nieser, U. Reitebuch, and K. Polthier. 2011. CubeCover—Parameterization of 3D Volumes. *Computer Graphics Forum* 30, 5 (2011), 1397–1406. https://doi.org/10.1111/j.1467-8659.2011.02014.x

Jorge Nocedal and Stephen Wright. 2006. *Penalty and Augmented Lagrangian Methods.* Springer Science & Business Media.

J.A. Norato, B.K. Bell, and D.A. Tortorelli. 2015. A geometry projection method for continuum-based topology optimization with discrete elements. *Computer Methods in Applied Mechanics and Engineering* 293 (2015), 306 – 327. https://doi.org/10.1016/j.cma.2015.05.005

Gun OYuce, Woodbury R, Peters B, and Sheikholeslami M. 2010. *Elements of Parametric Design.* Routledge.

Daniele Panozzo, Enrico Puppo, Marco Tarini, and Olga Sorkine-Hornung. 2014. Frame Fields: Anisotropic and Non-orthogonal Cross Fields. *ACM Trans. Graph.* 33, 4, Article 134 (July 2014), 11 pages. https://doi.org/10.1145/2601097.2601179

Pauli Pedersen. 1993. *Topology Optimization of Three-Dimensional Trusses.* Springer Netherlands, Dordrecht, 19–30. https://doi.org/10.1007/978-94-011-1804-0_2

Davide Pellis and Helmut Pottmann. 2018. Aligning principal stress and curvature directions. In *Advances in Architectural Geometry.* Klein Publishing Ltd, 34–53.

Jesús Pérez, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, José A. Canabal, Robert Sumner, and Miguel A. Otaduy. 2015. Design and Fabrication of Flexible Rod Meshes. *ACM Trans. Graph.* 34, 4, Article 138 (July 2015), 12 pages. https://doi.org/10.1145/2766998

M.H. Rasmussen and M. Stolpe. 2008. Global optimization of discrete truss topology design problems using a parallel cut-and-branch method. *Computers & Structures* 86, 13 (2008), 1527 – 1538. https://doi.org/10.1016/j.compstruc.2007.05.019

Nicolas Ray, Dmitry Sokolov, and Bruno Lévy. 2016. Practical 3D Frame Field Generation. *ACM Trans. Graph.* 35, 6, Article 233 (Nov. 2016), 9 pages. https://doi.org/10.1145/2980179.2982408

Christian Schumacher, Jonas Zehnder, and Moritz Bächer. 2018. Set-in-stone: Worst-case Optimization of Structures Weak in Tension. In *SIGGRAPH Asia 2018 Technical Papers (SIGGRAPH Asia '18).* ACM, New York, NY, USA, Article 252, 13 pages. https://doi.org/10.1145/3272127.3275085

Ole Sigmund, Niels Aage, and Erik Andreassen. 2016. On the (Non-)Optimality of Michell Structures. *Struct. Multidiscip. Optim.* 54, 2 (Aug. 2016), 361–373. https://doi.org/10.1007/s00158-016-1420-7

Justin Solomon, Amir Vaxman, and David Bommes. 2017. Boundary Element Octahedral Fields in Volumes. *ACM Trans. Graph.* 36, 3, Article 28 (May 2017), 16 pages. https://doi.org/10.1145/3065254

Mathias Stolpe and Krister Svanberg. 2003. Modelling topology optimization problems as linear mixed 0–1 programs. *Internat. J. Numer. Methods Engrg.* 57, 5 (2003), 723–739. https://doi.org/10.1002/nme.700

Kam-Ming Mark Tam, James R Coleman, Nicholas W Fine, and Caitlin T Mueller. 2015. Stress line additive manufacturing (SLAM) for 2.5-D shells. In *Proceedings of International Symposium on Shell and Spatial Structures.*

Kam-Ming Mark Tam and Caitlin T Mueller. 2017. Additive Manufacturing Along Principal Stress Lines. *3D Printing and Additive Manufacturing* 4, 2 (jun 2017), 63–81. https://doi.org/10.1089/3dp.2017.0001

Marco Tarini, Kai Hormann, Paolo Cignoni, and Claudio Montani. 2004. PolyCube-Maps. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 853–860. https://doi.org/10.1145/1015706.1015810

The MathWorks, Inc. 2018. MATLAB. https://mathworks.com/products/matlab.html

Erva Ulu, James Mccann, and Levent Burak Kara. 2017. Lightweight Structure Design Under Force Location Uncertainty. *ACM Trans. Graph.* 36, 4, Article 158 (July 2017), 13 pages. https://doi.org/10.1145/3072959.3073626

Nobuyuki Umetani and Ryan Schmidt. 2013. Cross-sectional Structural Analysis for 3D Printing Optimization. In *SIGGRAPH Asia 2013 Technical Briefs (SA '13).* ACM, New York, NY, USA, Article 5, 4 pages. https://doi.org/10.1145/2542355.2542361

Weiming Wang, Tuanfeng Y. Wang, Zhouwang Yang, Ligang Liu, Xin Tong, Weihua Tong, Jiansong Deng, Falai Chen, and Xiuping Liu. 2013. Cost-effective Printing of 3D Objects with Skin-frame Structures. *ACM Trans. Graph.* 32, 6, Article 177 (Nov. 2013), 10 pages. https://doi.org/10.1145/2508363.2508382

Li Wei, Zheng Anzong, You Lihua, Yang Xiaosong, Zhang Jianjun, and Liu Ligang. 2017. Rib Reinforced Shell Structure. *Computer Graphics Forum* 36, 7 (2017), 15–27. https://doi.org/10.1111/cgf.13268

Tomás Zegard and Glaucio H. Paulino. 2015. GRAND3 – Ground Structure Based Topology Optimization for Arbitrary 3D Domains Using MATLAB. *Struct. Multidiscip. Optim.* 52, 6 (Dec. 2015), 1161–1184. https://doi.org/10.1007/s00158-015-1284-2

Qingnan Zhou, Julian Panetta, and Denis Zorin. 2013. Worst-case Structural Analysis. *ACM Trans. Graph.* 32, 4, Article 137 (July 2013), 12 pages. https://doi.org/10.1145/2461912.2461967