Fabricable Flows for Topology Optimization

Sara Tang
1001145058

June 24, 2018

Abstract

We present a method of producing structually sound designs that can be assembled in a sequence of feasible steps. Traditionally, the topology optimization problem has been solved monolithically: however, in actuality structures are assembled not as a whole but in parts.

In our method, we partition the given structure into a set of "levels", which are then optimized cumulatively to obtain a sequence of sound designs. We present an objective function that gives preference to beams in previous stages of the design to ensure a minimal set-difference between optimized levels. We demonstrate our method on a variety of different structures of varying degrees of detail and stability.

1 Introduction

The ability to construct structurally sound structures is essential to (construction, models, engineering, I don't know Dave please halp). With the advent of consumer-friendly 3D printers, we are creating more models than ever (wow this sounds lame). But the size of such models are limited to the size of the printer, and building larger designs brings about headaches of falling structures and ohmygod I'll just write this later.

Talk about why it's important to build things in levels and not monolithically.

Talk about the different parts of the paper.

2 Background

Given a structure, presented as a set of vertices in 2- or 3-space and connecting bars, and a set of forces acting on the vertices, the traditional topology optimization aims to produce a superset of **sparse** and **possibly overconstrained** bars that are **structurally stable**.

At a high level, we do so by considering various deformations on our structure, calculating the internal force on each bar given the deformation, and attempting to balance the given external forces with the internal ones.

Consider the following structure in 2-space:

[INSERT IMAGE HERE LOL]

Let u_i be a 2-vector representing the **displacement** at vertex i. Let σ_{ij} be a 2-vector representing the **stress** on some bar connecting vertices i and j. Each bar has an internal force dependent on the amount of stress being applied on it, as well as an external force that is an input to the problem. A **structurally stable** design is one such that internal and external forces are balanced, that is, $f(\sigma) = f_{ext}$ for all bars.

2.1 Measuring stress

TODO: Consolidate when you use the vect sign and when you don't lol

Consider the following deformation of a single bar (x_1, x_2):

[INSERT PICTURE HERE.]

where $u = (\overrightarrow{u_1}, \overrightarrow{u_2})$ (SHOULD BE COLUMN VEC) is infinitessimally small in practice.

Define $A^T = [-I\ I]$ such that $\Delta u = vectu_1 - \overrightarrow{u_2} = A^T u$.

Let $\overrightarrow{v}$ be the unit direction from x_1 to x_2. We will only consider the deformation in direction $\overrightarrow{v}$, ie.

$$\overrightarrow{v} = \frac{x_2 - x_1}{\|x_2 - x_1\|}$$

As such, we project Δu onto $\overrightarrow{v}$, then multiply by Young's modulus to obtain the stress on (x_1, x_2) given deformation u:

$$\sigma := E v^T \Delta u \tag{1}$$

where E is Young's modulus.

2.2 Measuring internal force

Given a bar with stress σ, the internal force is a two-vector $f =$ (ANOTHER COLUMN VECTOR), where

$$\begin{aligned} f_1 &= \overrightarrow{n} \cdot \sigma \\ &= \overrightarrow{n} E v^T \Delta u \end{aligned}$$

Since $\overrightarrow{n}$ is exactly v,

$$\begin{aligned} &= v E v^T \Delta u \\ f_2 &= -v E v^T \Delta u. \end{aligned}$$

Thus,

$$f = \begin{bmatrix} I \\ -I \end{bmatrix} \left[vEv^T \Delta u \right] \tag{2}$$

2.3 All together now

We now have a way of deriving the internal force on a single bar given its deformation. We extend this notion to all bars in a given structure by defining the matrix K of dimensions $m \times n$ where m is the number of bars and n is the number of vertices such that, given a displacement of vertices u, Ku returns a m-vector representing the internal force on each bar.

Furthermore, there is this J matrix that I don't get, but basically $Ju = 0$, where

$$J = \begin{bmatrix} A^T v_1^T \\ A^T v_2^T \\ \vdots \\ A^T v_n^T \end{bmatrix}.$$

Thus, our goal is to solve u for the following constraints: $Ku = f$ and $Ju = 0$, that is

$$\begin{bmatrix} K & J^T \\ J & 0 \end{bmatrix} \begin{bmatrix} U \\ \lambda \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix} \tag{3}$$

where λ represents the tension on the bars.

Finally, we eliminate all bars where $\lambda = 0$, that is, bars that are no subject to tension or compression.

2.4 The Dual

Although the above formulation will return to us a structurally sound design, it will attempt to distribute the forces as evenly as possible, across all bars. We wish to do the opposite: distribute the forces as sparsely as possible across the bars, such that some bars will have no force on them and can thus be removed from the structure. To do so, we consider the dual of the above formulation. By using Gaussian elimination, we eliminate the U from Equation 3 to obtain:

$$-(JK^{-1}J^T)\lambda = J^T K^{-1} f \tag{4}$$

By representing λ as a difference of tensions and compressions, $\lambda_T - \lambda_C$, where $\lambda_T, \lambda_C \geq 0$, we can express sparsity as minimizing the L_1-norm, that is:

$$\|\lambda_T - \lambda_C\| = \sum \lambda_T + \sum \lambda_C$$

Thus, we obtain the following linear optimization problem:

$$
\begin{aligned}
\lambda_T^*, \lambda_C^* &= \arg\min \sum \lambda_T + \sum \lambda_C \\
\text{subject to} - (JK^{-1}J^T)(\lambda_T - \lambda_C) &= J^T K^{-1} f \\
\lambda_T, \lambda_C &\geq 0
\end{aligned}
\tag{5}
$$

which, in addition to the constraints expressed in Equation 3, also attempts to sparsify the tensions and compressions across all bars.

2.5 Overconstraining the problem

As an additional note, we may augment the original design with additional bars to ensure there is a structurally sound solution.

[PRESENT AN EXAMPLE.]

3 Implementation

3.1 Augmentation 1: Levels

Our first step into the unknown (lol) is to determine the substructure to be optimized at each step in our sequence. A simple and intuitive way of doing this is to label each bar with a level, from the ground up. This can be easily done with a breadth-first search starting at the ground vertices.

3.2 Augmentation 2: A better objective function

We may not wish to weight each bar the same. For example, we may prefer bars in the original structure over overconstrained ones. Thus, instead of simply summing up the tensions and compression, we may assign a larger weight to tensions and compressions corresponding to overconstrained bars and a smaller (or even 0) weight to tensions and compression corresponding to original ones.

4 Results

5 Conclusion and Future Work