SHAPE PROGRAMMING & SELF-ASSEMBLY OF BEAD STRUCTURES

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science

by

Cameron Nelson

August 2021

ABSTRACT

This paper demonstrates the potential of a robust, low-cost approach to programmable matter using beads and string to achieve complex shapes with novel self-organizing and deformational properties. The method is inspired by the observation that beads forced together along a string will become constrained until they form a uniform rigid shape. This behavior is easily observed in any household string and flat-faced beads and recalls the mechanism behind classic crafts such as push puppets. However, specific examples of architectural applications are lacking. We analyze how this phenomenon occurs through static force analyses, physical tests, and simulation, using a rigid body physics engine to validate digital prototypes. We develop a method of designing custom bead geometries able to be produced via generic 3d printing technology, as well as a computational path-planning toolkit for designing ways of threading beads together. We demonstrate how these custom bead geometries and threading paths influence the acquired structure and its assembly. Finally, we explore a means of scaling up this phenomenon by fabricating in carveable foam, suggesting potential applications in deployable architecture, mortarless assembly of non-funicular masonry, and responsive architectural systems.

BIOGRAPHICAL SKETCH

Cameron Nelson holds a Bachelor’s of Arts in Mathematics and Architecture from Yale University, where they received the Harvey Geiger Fellowship for travel in architecture. There they also completed an independent study on the topic of topology optimization in the context of the Polybrick 2.0 project from Jenny Sabin Lab. At Cornell they served as a research assistant conducting software development for the Circular Construction Lab, and taught as an assistant in several courses including Generative Design and the Matter Design Computation core studios. Their research interests include optimization, geometry, graph theory, and heuristics.

Dedicated to the dreamers of the day

ACKNOWLEDGMENTS

I wish to acknowledge my thesis committee chair, Professor Jenny Sabin, and minor members, Professors Yasser Gowayed, and Craig J. Fennie, for their mentorship and inspiration.

I wish to further thank Jenny for pioneering the Matter Design Computation research degree program at Cornell University. I feel honored to have been a part of its inception. Thank you for seeing potential in me.

I thank my parents and my brother for endless patience, support, and laughter.

To my partner Mahshid, I am so grateful for our connection and look forward to lifelong collaboration, ghorboonet beram.

To my partner Dawn, thank you for witnessing me and helping me grow. Life with you has been a dream.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Beads, as an art form, are closely entwined with the story of human culture, dating back at least 100,000 years. Over the millennia their uses have ranged from jewelry, to currency, to game tokens, to ritual objects. Beyond aesthetic or symbolic meaning, they also have had physical applications as insulators, rope stoppers, mechanical gears, rollers, and weights. One familiar example of a mechanical bead assembly is "push puppets" or "thumb toys," an example of which is shown in Figure 1. These typically consist of a figure, often an animal or humanoid whose limbs are made up of discrete beads along a string, atop a small platform with a button concealed beneath. When the button is pressed, the string within the beads slackens, and the previously rigid figure collapses. At the release of the button, tautness returns to the string and the figure stands back up. While this mechanism is often utilized in toys as a mere curiosity, this thesis will explore its practical application to structures, and particularly its properties of (self-) assembly.

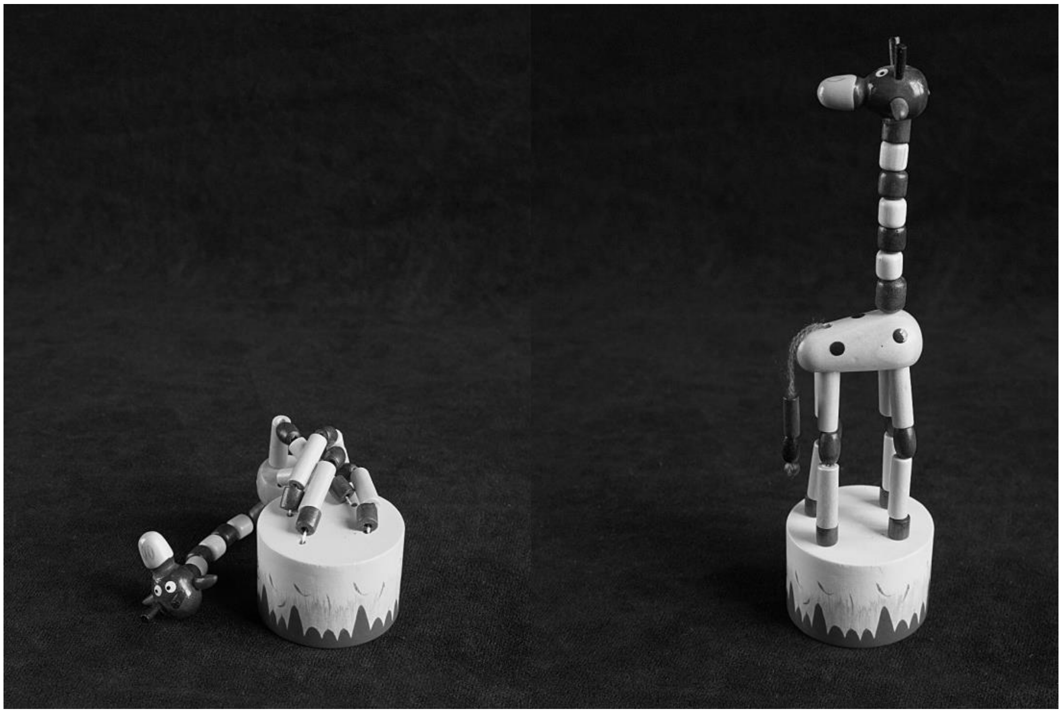


Figure 1. A push puppet with button pressed and internal strings slackened (left); with button released and strings taut (right). (http://de.wikipedia.org/wiki/Benutzer:Drahkrub, 2010)

If beads can be formed into controllably rigid shapes resembling animals and humans, we hypothesize they can be coaxed into more architectural forms such as shell structures, awnings, tent frames, and the like. Moreover, we can extend the mechanism by manipulating the shape of the beads and the sequence of their threading to enhance these shape-finding properties. For example, when two cylindrical beads are forced together along a string, their flat faces will align and they will become one rigid cylinder, but if one or both of these beads were mitered like a cylinder sliced by a plane at an angle, then the assemblage of the two beads would form a predictable angle. The combination of many such beads in one or more interconnected sequences would offer even greater shape control than is seen in push puppets.

Basic Definitions

We considered beads to be solids with one or more hollow channels that may or may not intersect. This allows Y- and X-shaped beads (Figure 3), as well as more complicated topologies. One of the most important aspects is the geometry of the contact surfaces, which determines orientation constraints and interlocking behavior as well as the torque between adjacent beads. For example, generic flat interfaces constrain position but not relative rotation, whereas interlocking interfaces can partially or fully constrain adjacent beads. For complex geometries our design process begins from a skeletonized “framework”: a connectivity graph with nodes embedded in 3d space. This is so as to be as generic as possible. Our choice of framework is essentially arbitrary, and need not necessarily reflect any kind of pre-optimization or force analysis. For example, the geometry in Figure 2 is not a true funicular dome, and hence cannot stand without a tensile string holding the beads in compression. It would immediately collapse as a pure, mortarless masonry grid shell. Conceptually, the emphasis is not upon static form-finding to produce an ideal geometry at time of manufacturing, but rather to produce a robust assembly that will weather unpredictable loading conditions and still return to its original state. To generate manufacturable bead geometries, we apply a method we term "graph augmentation:" given a framework, we build volumetric beads about the edges such that the interface between two adjacent beads is aligned to the perpendicular plane of the edge between them in the framework (Figure 3).

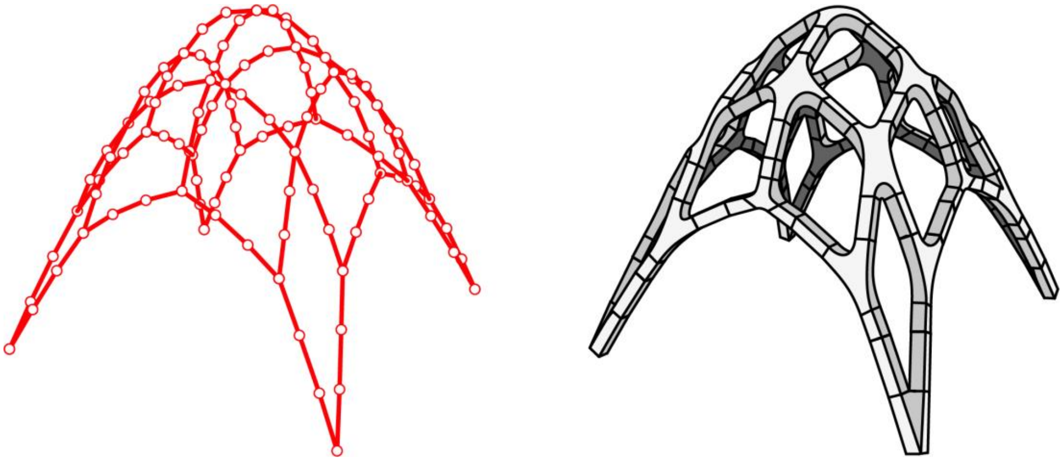


Figure 2. A complete generic dome framework, and the 3d printable bead geometries generated via graph augmentation.

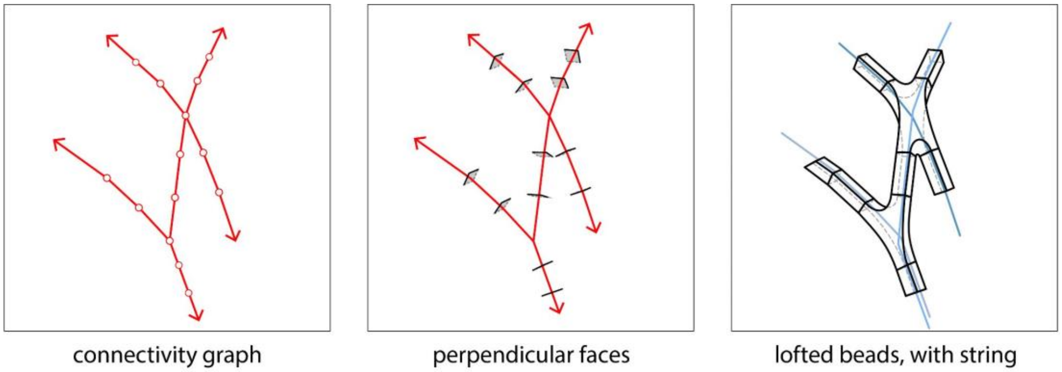


Figure 3. Graph Augmentation process. Left-to-right: framework, interfaces, and full solid geometry.

The channels within a bead are the topological holes, distinct from the cord itself which passes through them. For instance, Figure 3.c. shows a bead with an X-shaped internal channel and two cords passing through it, one from lower left to upper right, and the other from upper left to lower right. It could just as well have been threaded with two cords, one passing from lower left to lower right and the other from upper left to upper right, or any other combination, without changing the underlying topology of the node. See chapter 5 for a discussion of threading.

Precedents

To give an idea of the potential of such a system for architecture, specifically, we briefly describe several notable precedents below. While none of these examples exhibits all the traits of what we term "bead structures", together they form a picture of possible use-cases and properties of such assemblages.

A simple yet pertinent precedent is the "Self-Folding Protein" by the MIT Self-Assembly Lab.[1] This 1-dimensional sequence of thin cylindrical segments connected by hinges was modeled after the Crambin protein. The hinges allow rotation about a fixed axis up to some maximum angle, determined by the angle of the miter of the end faces of these segments. These hinges naturally tend to bend to their maximum extent, since at any other angle they are susceptible to swinging in either direction. Intuitively, if one were to randomly jumble and shuffle such a chain, gradually the straight chain would kink and wrinkle into a zigzag shape entirely determined by the sequence of miter angles of the segments, effectively encoding global shape via a series of part-to-part rules. An unordered form of agitation (random shaking) results in an ordered outcome (a predetermined global shape).

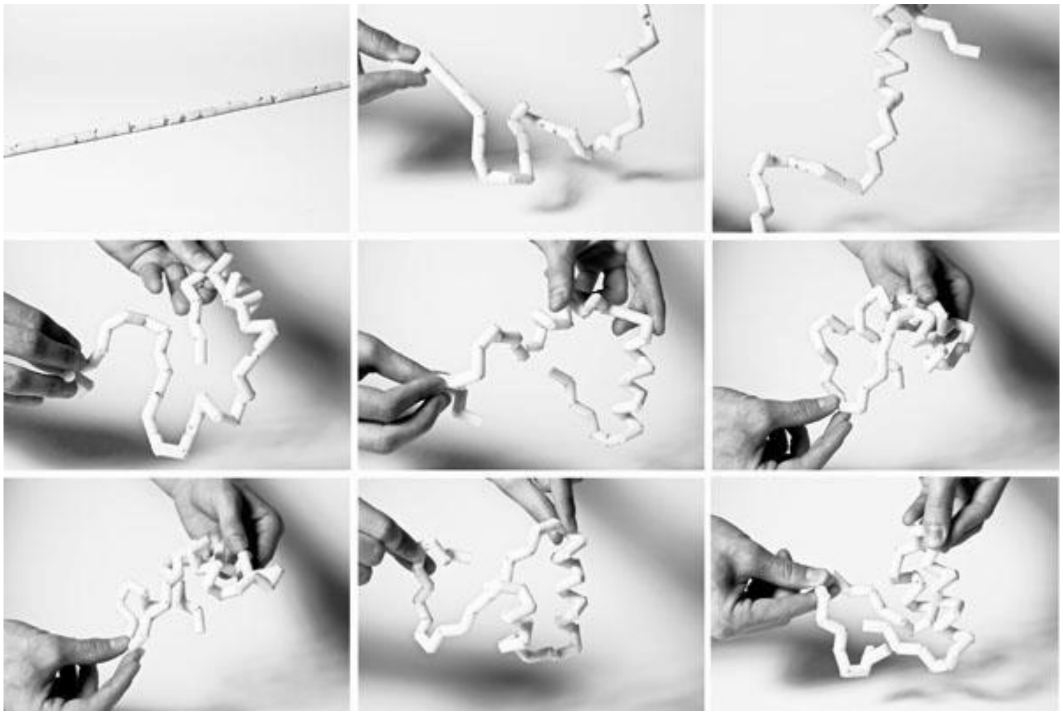


Figure 4. Self-Folding Proteins. Project by MIT Self-Assembly Lab
(Image Source: MIT Self-Assembly Lab) ("Self-Folding Proteins — Self-Assembly Lab" 2021)

As we will give special consideration to the self-assembly properties of bead structures, it is fitting that MIT's Self-Assembly Lab should furnish several precedents. Self-assembly is a suggestive term used with varying definitions in the discipline of architecture. Inevitably there is some input energy, and even physical or biological definitions allow that this force may come from without the system itself, subtly augmenting the meaning of "self" -organizing. For example, in the project "Fluid Crystallization" a wave chamber is agitated to ensure intermixing of the floating magnetic modules (Tibbits 2014). Therefore, we adopt the interpretation that self-organization indicates an unordered form of agitation that nevertheless yields an ordered result.

A similar project from Tibbits, the "Macrobot", also globally programs shape from a 1d sequence of local orientations. In this case, the segments are not cylinders, but rather truncated bi-cones formed of sheet-metal. The project, produced at the Center for Bits and Atoms under a DARPA grant for Programmable Matter, explores motorized control of these angles to transform a single chain into numerous shapes. Our approach differs in that our bead assemblies are non-motorized, with no electronic componentry whatsoever. Instead, all control is at the level of shape and basic mechanical actuation, such as pulling a string. This simplicity not only adds to the robustness of the system; it also allows us to focus more on the geometric aspects of programmability of matter.

Being produced of lightweight, stiff materials held in compression by a structurally separate tension member, our bead assemblies have much in common with post-tensioned masonry. A particularly relevant project is the "Periscope" by Matter Design Studio, which used a single sequence of large, robotically wire-cut polystyrene foam blocks to achieve a stereotomic tower 60 feet in height on a footprint of less than ten feet square (Clifford and McGee 2011). Here, the tensioning cables are external to the structure, acting as guy lines, but the principle is the same even if the force lines were constrained to pass through the volume of the beads or segments. In chapter 8 we explore scaling up the phenomena discussed in the start of this thesis through a similar material system, employing polystyrene foam beads with internal channels. Moreover, we seek to extend this example by applying tension dynamically, taking full advantage of the shape-changing qualities of flexural materials in addition to the structural qualities of rigid ones. These dynamic properties suggest precedents in deployable and tensegrity architecture, although bead structures have the property of distributing compressive forces externally and tension internally, which contrasts with key examples such as inflatables and the iconic bar-and-cable systems of Kenneth Snelson and Buckminster Fuller.

A project by the Digital Structures Lab at MIT exploring post-tensioning of non-funicular structures offers an interesting comparison with these bead systems. The project makes use of graphic statics, a centuries-old method of relating form to force through vector diagrams, to find the position and stress of a post-tensioning cable that could compensate for non-axial compressive forces. Graphic statics has regained popularity in architecture in the last decade, in part thanks to the contributions of the Block Research Group at ETH Zurich and the Polyhedral Structures Lab at the University of Pennsylvania. The Funicular Explorations Pavilion by the Digital Structures Lab offers a demonstration of their method applied to a wall-leaning half-arch. The arch is mortarless and nonfunicular, but it is held in equilibrium by external tension wires, in a manner similar to the arches of the Pavilion of the Future engineered by Peter Rice of Ove Arup & Partners for the 1992 Universal Exposition in Seville. Bead structures exhibit similar versatility; bead masonry need not be funicular, because the tensile cord within it carries non-compressive loads required to stabilize the structure. But unlike externally post-tensioned masonry, no calculation is required. In theory the structure will be stable for some (possibly very large) applied force, which can be found simply by increasing the tension until the structure stands. Moreover, while the method of external post-tensioning via graphic statics becomes unwieldy for 3d structures, the method of beads extends naturally to the third dimension.

CHAPTER 2
PARAMETRIC INTERPRETATION

We begin to analyze the mechanism of rigidifying beads by examining the equilibria and the configuration spaces of a variety of common bead geometries. By "equilibria" we refer to the relative position of two adjacent beads when pulling on the string would not induce the beads to move with respect to one another. One could also think of this as a "locked" or "jammed" state. "Configuration space" refers to the set of valid tuples of parameters of a system. It is therefore inherently dependent on which parameters we select for analysis. Depending on the geometry in question, it can be difficult to define the configuration space and explore all possible jammed states, but we can construct several demonstrative examples.

Cylindrical Beads

One of the most common primitive bead geometries is a cylinder with a hollow channel through its axis of symmetry. Our definition of self-assembly refers to the emergence of "order", which for the sake of argument can be quantified as the size of the system's configuration space, that is, the volume spanned by the valid domains in its parameter space. For example, we could measure the maximum angle between consecutive cylindrical beads with respect to the distance separating their facing ends. It is helpful to consider the channel to have effectively zero radius, and to assume the string is taut, such that the cord forms a piecewise straight curve between the centers of the two beads' adjacent flat faces. If we define the length of this cord as l, the radius of the beads as r, and the maximum angle between the two beads' axes as θ_{max}, we obtain the set of equations below whose plot is shown in Figure 5, revealing a continuous monotonic decrease in θ_{max} as l decreases. In practice, this means that any shortening of the cord, such as by tension, will reduce the size of the configuration space until the system is forced into a constrained shape.

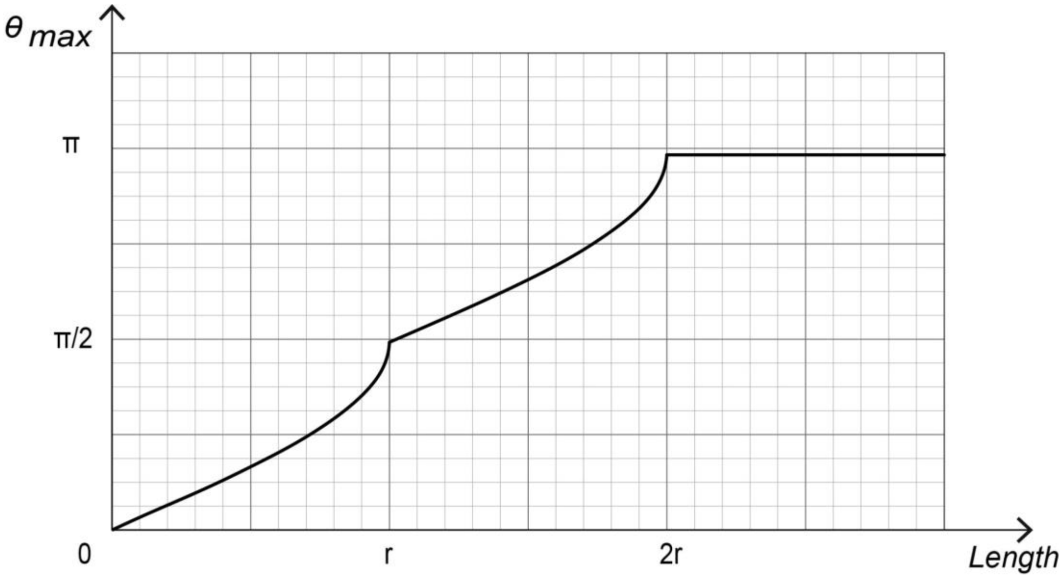


Figure 5. Maximum angle attainable between cylindrical beads as a function of the length of cord separating them. Note that $\theta_{max} \to 0$ as $l \to 0$, continuously but with discontinuous first derivative at $l = 2r$ and $l = r$. This graph illustrates the shrinking of the configuration space with the tensioning of the system.

This graph also suggests something about the cylindrical bead's jammed states. Consider the singularities at $l = 0$, r, and $2r$. The first and most obvious equilibrium is when these beads are as close to each other as possible along the string between them, so that their adjacent flat faces coincide. Then the two flat circular faces are pressed against one another, with the string itself resisting any motion parallel to the surfaces, so that the forces compressing the beads together cancel one another. This is what we observe when a series of short cylindrical beads along a string become like one long rigid rod after they are compressed. It is represented by the point $l = 0$, and visualized in Figure 6.a.

We can also think of two beads in contact as a lever, with their point of contact forming the fulcrum. Further jammed states often occur when the string's pulling force acts along a line passing through the fulcrum; since the moment of one bead with respect to the other is determined by the perpendicular distance of the force line from the fulcrum, there will be no relative rotation. Practically speaking, even when the force line merely passes close to the fulcrum, the mechanical effort to dislodge the near-locked state may still be infeasible. At $l = r$ and $l = 2r$ the beads can take on the configuration in Figure 6.b. and Figure 6.c., respectively. In these figures the fulcrum is highlighted. Note that it is indeed transected by the line of the cord.

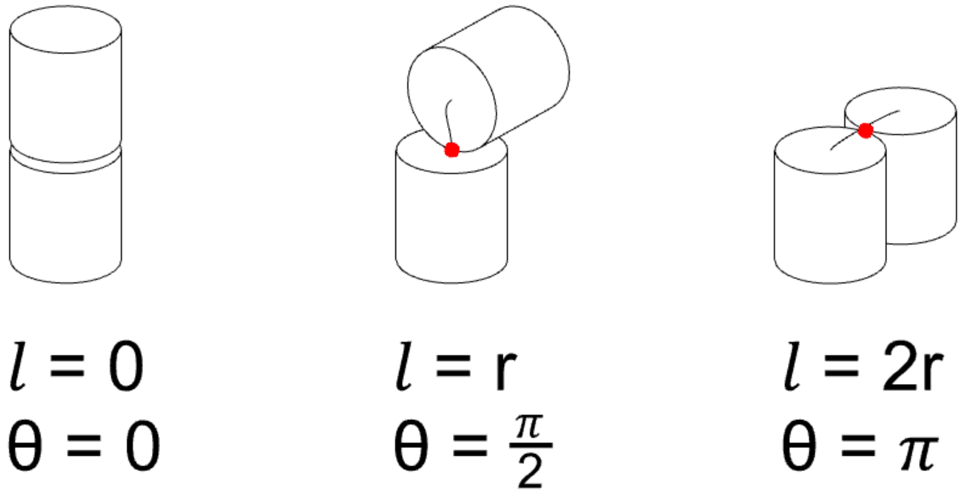


Figure 6: locked states, solid cylinder

Tubular Beads

We assumed until now that the channel's radius is much thinner than the outer radius of the beads. When this is not the case, beads' equilibria are not as predictable. For example, if the inner radius is approximately equal to the outer radius, i.e. if the tube wall has effectively zero thickness, then the probability that they line up exactly is also basically zero. A slight displacement of one tube perpendicular to its axis will reduce the points of contact between the beads to just two, and they will be free to rotate about the line between these two points up to some maximum angle as demonstrated in Figure 7. Even if the wall thickness is nonzero, beads can rotate around the line between the points where the outer edge of one tube intersects the inner edge of the other. If more than two tubular beads meet at a point, it becomes even less likely they will align axially, as seen in Figure 8. As such we often rely on topology programming rather than shape programming to induce rigidity in a system of tubular beads. For bead systems like these the analysis of mathematical rigidity of the overall network becomes especially applicable. See chapter 3 for a continuation of this discussion.

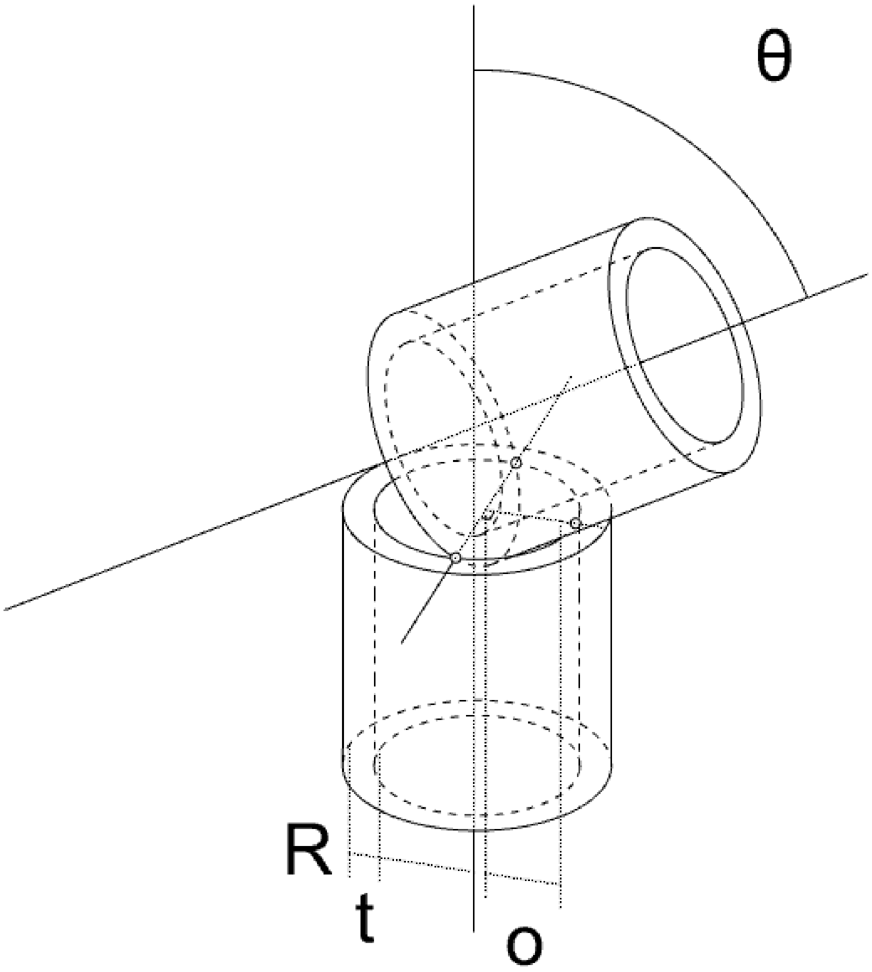


Figure 7. The angle ***θ*** *of three-point contact can be determined for a given outer radius R, wall thickness t, and axial offset o of one bead with respect to the next.*

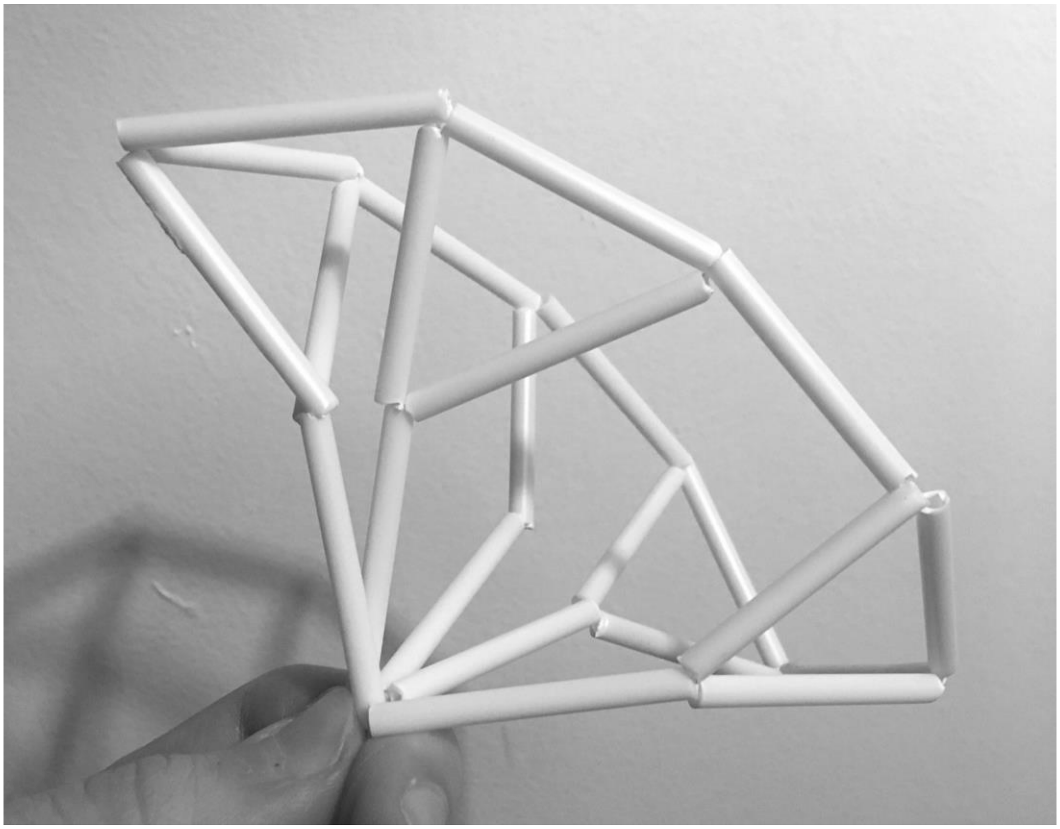


Figure 8. A nonrigid framework with degrees of freedom partially reduced through greater topological connectivity.

Spherical Beads

Spherical beads have only one equilibrium: an attractive force between the endpoints of these beads will always bring the beads' endpoints closer together, until they are coincident. This can be demonstrated by contrast with the cylindrical example. The cylinder's locked states occur when the thread between two beads passes through the fulcrum, i.e. the point of contact around which they are rotating. But if a line passes through two points on the surface of a sphere, one of them being the endpoint and the other the point of contact, then the only way it will not intersect the interior of the sphere is if those two points are coincident, i.e. the endpoint *is* the point of contact. This explanation generalizes to any convex bead with smooth derivatives. This suggests that spherical beads are less likely to become jammed, and to an extent this is true; however, the lack of a flat interface means they touch in at most one point, which reduces the effective rigidity of the contact. Still, their interface locally approximates a flat plane, which can compensate slightly.

Bi-cone beads, another common shape in beadcraft and the shape chosen for the units of the Macrobot project by the MIT Self-Assembly Lab, demonstrate this even more clearly: even when the endpoints of two adjacent beads are coincident, they are free to rotate around that point up to some maximum angle. This can be used to create a constraint on the angle between two adjacent beads, without constraining the plane in which that angle lies so that it can rotate freely around the axis of either bead.

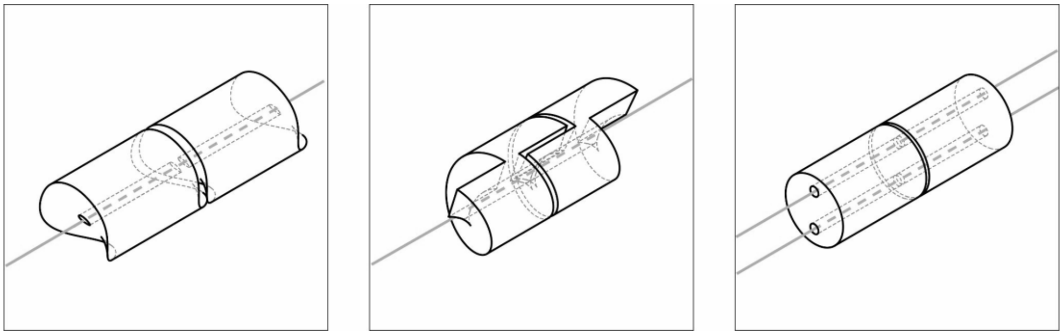


Figure 9. Three common types of orientation-constraining interfaces, with different embedded intelligences: interlocking, screw-in, and multi-cord interfaces.

Interlocking Beads

Figure 9.a. shows a bead with a simple interlocking interface. This helps secure the rigidified beads against becoming dislodged, but if the interlock is too specific it can be hard to coax the beads into their precise interlocking positions. To give an intuitive example, consider beads with a conical key of a certain angle τ. The bigger τ, the blunter the point of the cone; the smaller, the pointier. We might correctly guess that a pointier cone will fit more securely in an adjacent bead, but will be more difficult to line up in that position in the first place, like threading a needle. We can verify this by observing the equations for the maximum angle between adjacent beads with a given τ. R is derived from τ and represents the length from the tip of the cone to its circumference. These equations become the same as the above equations for cylindrical beads in the limit as τ goes to $\pi/2$.

However, for τ < π/2, there is a discontinuity in the graph at *l* = R. This manifests as a vertical tear in the graph of the configuration space, or physically as a popping or flipping action of the beads, where their relative angle suddenly becomes much more constrained or much freer with only a small change in the length of the cord between them.

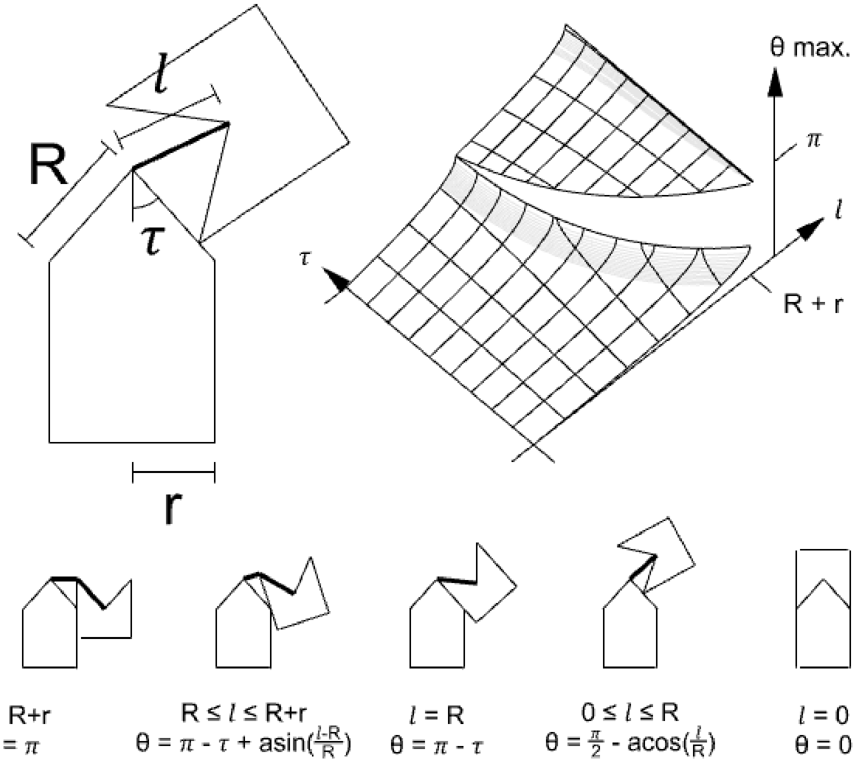

l
R
τ
r
θ max.
π
l
τ
R + r
l = R+r
θ = π
R ≤ l ≤ R+r
$\theta = \pi - \tau + \mathrm{asin}(\frac{l-R}{R})$
l = R
θ = π - τ
0 ≤ l ≤ R
$\theta = \frac{\pi}{2} - \mathrm{acos}(\frac{l}{R})$
l = 0
θ = 0

Figure 10. Configuration space of conical interlocking beads, shown in section. The 3d graph of the configuration space boundary is discontinuous for τ < π/2.

CHAPTER 3
RIGIDITY INTERPRETATION

Up to this point we have considered locked states from a stereotomic standpoint. We have asked whether systems of forces on solid objects would induce relative motion. We could also approach questions of rigidity from a purely topological perspective. Given a network of incompressible and inextensible “bars” in space, connected at a collection of nodes about which they are free to pivot, we call this network rigid if the only movements of the nodes that preserve the lengths of all the bars are the global translations and rotations of the entire network: so-called rigid motions []. This is easily verified using the rigidity matrix of the network, defined as follows: if the network is embedded in d-dimensional space, with n nodes and m edges between them, then we construct a matrix with m rows and $n*d$ columns wherein the $(3*v+i)^{th}$ entry of row e is 0 if node v is not an endpoint of edge e, and otherwise is the difference between the i^{th} coordinates of v and the i^{th} coordinate of the opposite endpoint, u (Roth and Whiteley 1981). An infinitesimal motion is a set of n velocity vectors, one for each node in the framework, or, equivalently, the $n*d$-dimensional vector formed by concatenating them into one. In the latter case, observe that the inner product of a motion with any given row of the rigidity matrix is 0 if the instantaneous change in length of the corresponding edge is 0 under that motion. If $L(e)$ denotes the length of e:

$$dL(e)/dt = (v-u) \cdot du/dt + (u-v) \cdot dv/dt$$

i.e., the instantaneous change of $L(e)$ is the speed of v towards u, plus that of u towards v, which is equivalent to the inner product of the motion vector with row e of the rigidity matrix. Thus, the nullspace of the rigidity matrix represents the space of all length-preserving motions.

One way of applying this to our case is to abstract beads as straight lines. These lines have some fixed length that cannot change, so we can think of them as bars in a framework, and the strings between them act as arbitrarily long inextensible cables. This is especially appropriate for beads with a high ratio of length to radial extent, such as tubular members of a space frame. Jammed or “rigid” states are represented by sets of positions of the nodes of the network satisfying that the only valid infinitesimal motions are nontrivial. By the above observation that the inner product of rows with a motion is 0 if the corresponding edge length does not change, we can determine whether a certain arrangement is rigid by computing the nullspace of the rigidity matrix. We wrote a custom C# script for the Grasshopper parametric CAD environment that takes as input a set of 3d line segments defining a network and outputs a boolean value indicating the rigidity of the network by comparing the nullity of the rigidity matrix with the rank of the set of rigid motions, which is (d+1)-choose-2 = 12 in 3 dimensions. The component outputs true if the former is greater than the latter. How *much* greater indicates the number of the network’s nontrivial degrees of freedom, giving a quantitative sense of how nonrigid the system is.

Integrating an infinitesimal motion will theoretically yield a new valid embedding. To find a nontrivial motion of interest, we can take any random n*d-vector in the nullspace and successively subtract from it its components along the directions of the rigid motions: three translations and three rotations. Once the rigid motions are factored out, we move all nodes some small multiple of their corresponding velocity vector, then use dynamic relaxation to correct any small changes in the edge lengths that might have been caused by the linearity assumption. We can then recalculate a basis of the nullspace of the rigidity matrix, project the previous motion onto this space to find its closest valid successor, then repeat the above steps. Currently our implementation of this pseudocode is not optimized and runs too slowly to be of aid in design, but this could be an interesting direction for future work, particularly if it were to help in visualizing the boundaries of the regions of valid motions in n*d dimensions.

A given topological network not embedded into space can have multiple rigid embeddings, possibly even with the same edge lengths. However, given two embeddings with equal edge lengths it is not guaranteed that one can morph into the other. The configuration space of such a network is the subset of *n***d*-dimensional space which represents valid embeddings, and this space may actually be concave or even a disjoint collection of regions.

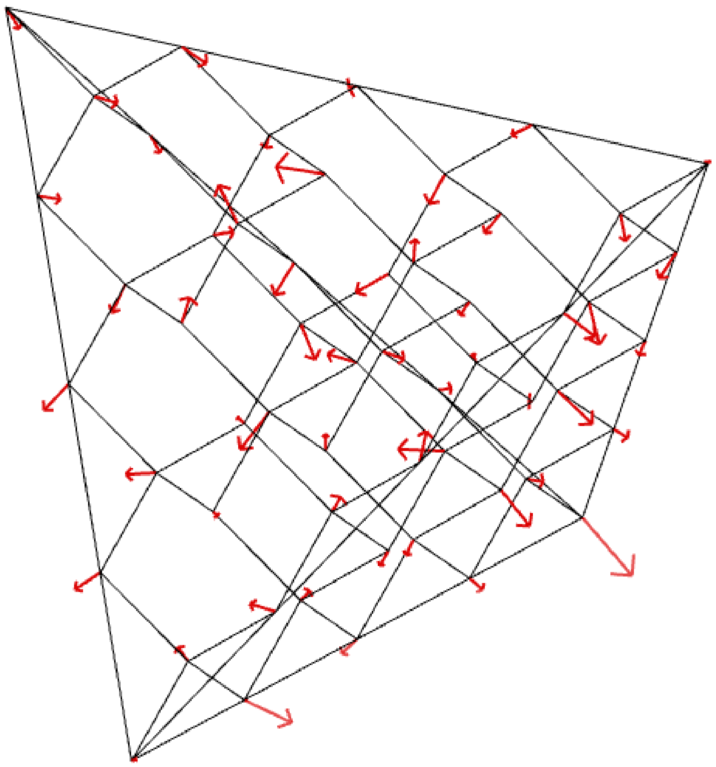


Figure 11. Visualization of a nontrivial infinitesimal motion of a complex framework.

We are also interested in how we arrive at such rigid states from a loose starting configuration, only permitting beads to move closer together, i.e. only permitting the strings between them to get shorter. This would imply that the inner product of the infinitesimal motion with a given row is 0 if that row corresponds to a bar, or simply nonnegative if the row corresponds to a cable. These are all the nonrigid motions that are mapped to strictly positive vectors in *m*-dimensional space by the rigidity matrix.

The Grasshopper environment makes it easy to apply the above techniques to visualizing an infinitesimal motion as a discrete vector field over the nodes of a framework, as in Figure 11. Further research might investigate visualizing the configuration space of valid embeddings in order to identify which embeddings can smoothly morph from one to the other, and what is the shortest path through the parameter space between them. The primary challenge would be the configuration space's high dimensionality, but common methods like projection, principal component analysis, and autoencoders might prove useful to break the question into manageable parts.

It is important, however, to note some of the shortcomings of the rigidity method. The most obvious is the fact that collisions are not accounted for, so bars and cables can pass through one another while integrating an infinitesimal motion. There are ways around this problem, but all require additional computational complexity to be inserted into the algorithms.

CHAPTER 4
PHYSICAL INTERPRETATION

In theory, using infinitely strong and frictionless materials, one can create a self-organizing bead structure approximating any given framework. In practice, however, friction, strength, and geometry play a role that can be limiting or fruitful.

Consider the radius as the maximum distance from a point of contact between two beads to the string where it passes through them. A greater radius reduces the required tension force by supplying more torque. However, assuming constant density and cross-section, the aspect ratio of radius to length is unimportant. Figure 12 illustrates that *n* beads each *l*/*n* in length will require the same force to rigidify as *m* beads each *l*/*m* in length, for any *n*, *m* > 0.

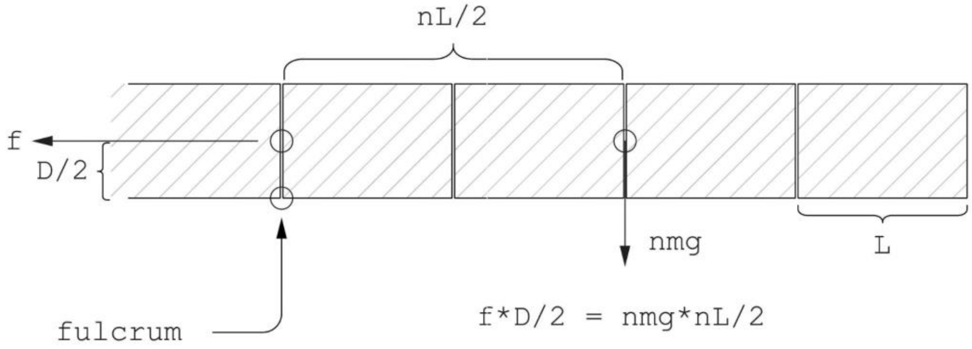


Figure 12. The figure assumes a cantilever subjected to gravity. Bead diameter D = 2r is inversely related to required tension f. L represents the bead length, m the mass per bead, and n the number of beads being supported. Because n is inversely proportional to both m and L, f is constant for changing n.

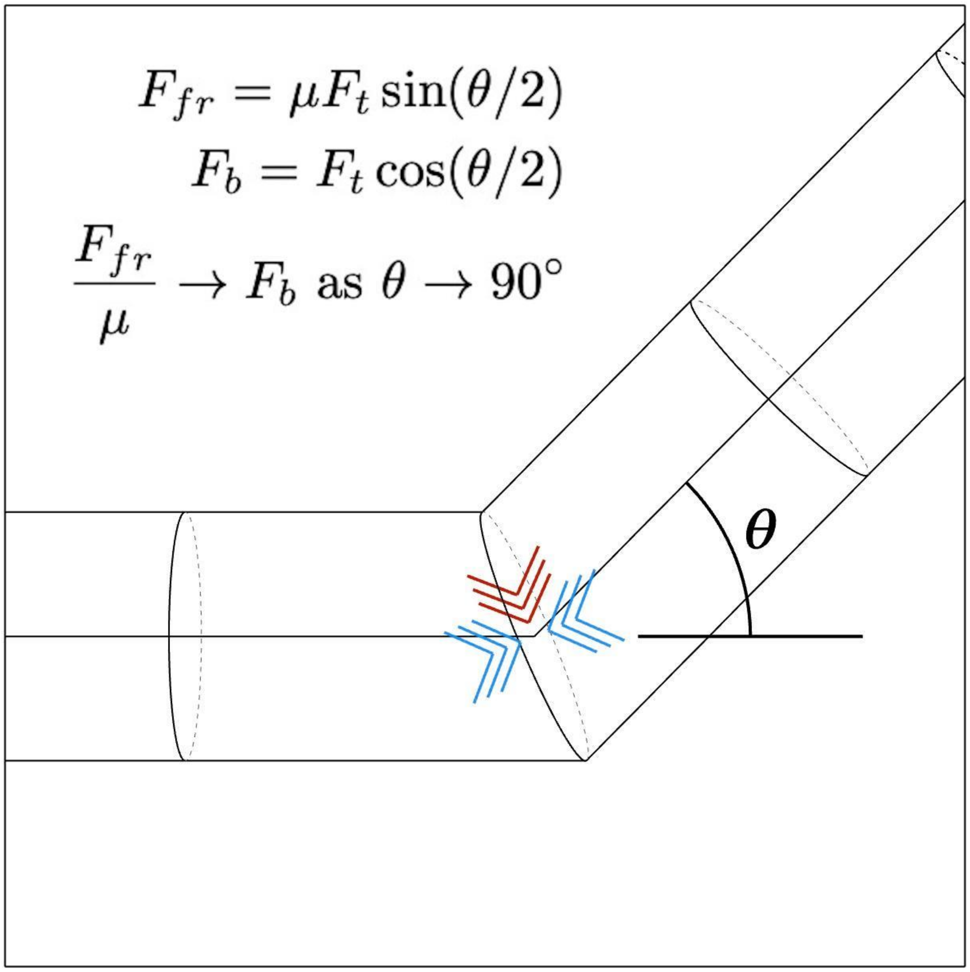

$F_{fr} = \mu F_t \sin(\theta/2)$
$F_b = F_t \cos(\theta/2)$
$\frac{F_{fr}}{\mu} \to F_b$ as $\theta \to 90°$
θ

Figure 13. Two beads being compressed along a cord, demonstrating the effect of miter angle on friction between bead and cord. If θ is the supplement of the miter angle, $F_b = F_t \cos(\theta/2)$ and $F_{fr} = F_t \sin(\theta/2)$. For example, if $\theta = \pi/2$, we would expect $F_{fr} \approx 71\% \mu F_t$.

Aspect Ratio

One hundred cylindrical beads each a centimeter in length will require the same force to rigidify as would ten beads each a decimeter in length. However, there are advantages to one or the other. For example, in reality the channel through these beads may not be exactly the same diameter as the string, allowing the beads some small degree of movement perpendicular to their axes. The accumulation of error from one bead to the next may produce large global displacements between the ends of the bead chain, and these become more pronounced the more beads there are in a sequence. Fewer components also makes assembly much faster. On the other hand, the more highly discretized chain of one hundred beads can potentially be packed into a smaller area when not rigid.

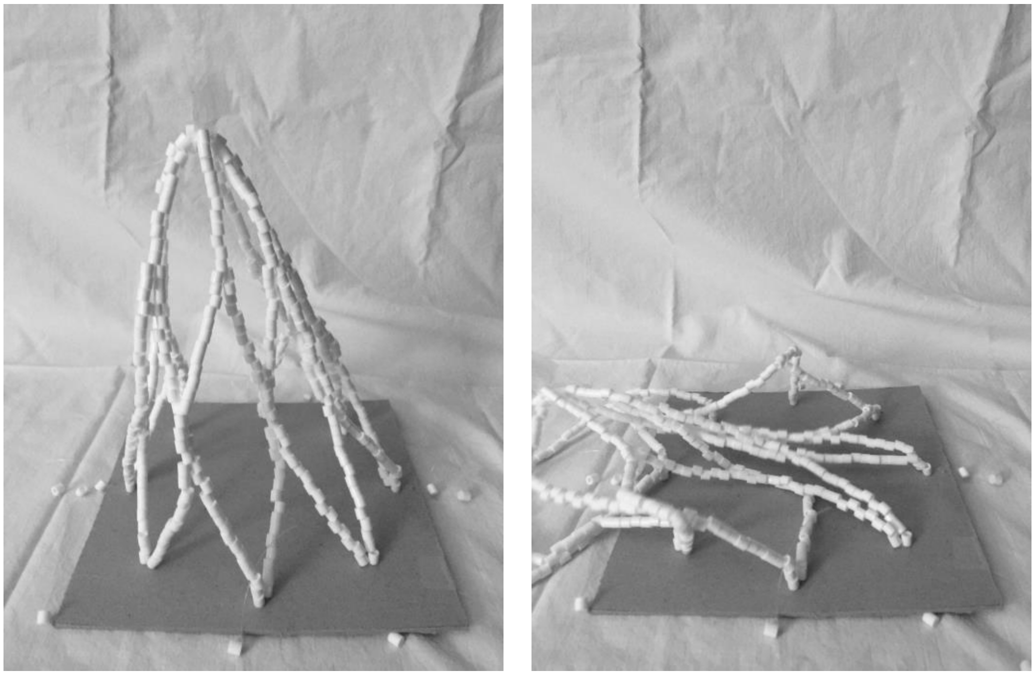


Figure 14. A low aspect ratio, or a highly discretized structure, must combat accumulation of error at its many joints. Here we also see the imprecision of tubular beads as compared with true solid cylinders.

Friction can significantly affect the transfer of tension. We observed this behavior even in relatively low-friction systems like the PLA and cotton-cord dome prototypes. In mitered beads like these, unlike in a straight chain of beads, the tension force applied to the cord is differently distributed, as seen in Figure 13. As the miter angle decreases, the normal force F_b between beads decreases, and the friction force F_{fr} between the string and the beads increases, requiring considerably greater force to pull.

CHAPTER 5
THREADING

For any network topology more complex than a single row of beads in series, there may be many ways to thread them such that a cord passes through each bead at least once. For instance, we might use several threads each following different paths, and some beads might be threaded through with multiple cords. There are many properties of a given threading path we might wish to optimize; here we consider (I) material economy, i.e. the length of cord used; (II) ease of assembly, particularly as relates to symmetry and standardized patterns; (III) shape control through topological constraint; and (IV) mechanical effort, especially as it relates to the tortuosity of the threading path.

Length

When the objective is to minimize the length of a single string, this reduces to the well-known Route Inspection problem (Edmonds and Johnson 1973). We implemented an algorithm using linear programming in the Python module PuLP (Mitchell and Dunning 2011) using an approach similar to the simulation pipeline: a dictionary-type object records the edges of a framework's underlying graph as pairs of indices of the vertices at their endpoints, and is then exported to a comma-separated value file that can be read by a Python script to exactly solve this problem. However, the threading paths produced by the Route Inspection algorithm can be highly asymmetric. This solution easily extends to *n*-cord threadings: minimizing the length of string when each bead interface must be traversed at least *n* times is identical to the single-cord case, only we duplicate each edge in the graph *n* times.

There are several other possible criteria that may be worthwhile subjects of further research. One might wish to minimize the total length of multiple independent paths when no path is longer than some maximum length. One may also only allow the cord to terminate at specific nodes, such as the feet of the dome in some of our physical models discussed later in chapter 7.

Symmetry

Often, when assembling a system by hand, a symmetrical threading path can be more intuitive to follow. The symmetries of a network are encoded in its automorphism group, a collection of ways of rearranging nodes of the graph such that path continuity is always preserved. By performing any such rearrangement on the vertices of a path, we obtain a symmetric copy. The challenge then is to determine a set of one or more paths whose symmetric copies will cover the edges of the graph with the fewest overlaps. For example, in the threading diagram in Figure 15 there are only two unique paths that have been transformed into eight copies by a subgroup of the graph's symmetries-- in this case, the mirror symmetries. Unlike true mirror symmetry, though, graph automorphisms are topological in nature, making them robust to warping. This more generalizable approach to path planning can identify redundancy even in geometries that don't bear obvious symmetry.

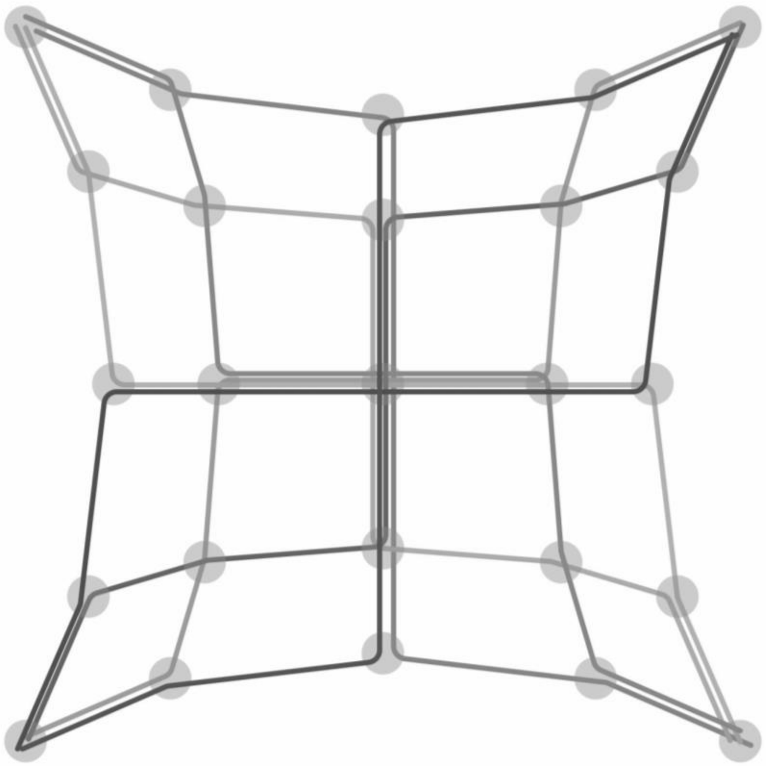


Figure 15. A diagram of the threading path applied to the initial dome prototype (Figure 2). Here, symmetry is emphasized.

Constraints

Geometric constraints upon one bead with respect to its neighbors can come not only from the bead's shape, but also potentially from the threading pattern. As mentioned above and shown in Figure 9, multi-thread systems offer the benefit of constraining rotations. Two strings will fully constrain adjacent beads with planar interfaces, and three strings will fully constrain arbitrarily-shaped beads, in the same sense that three points of contact are required to stabilize a chair. The dome model shown in Figure 25 exhibits a 2-cord system with planar interfaces. A different approach to multi-cord threading patterns can be seen in the tiled tube prototypes in Figure 16. Here each bead represents a node of degree 4 in the model's graph representation. Thus, with a minimum of four points of contact per bead the structure is fully constrained in position and orientation.

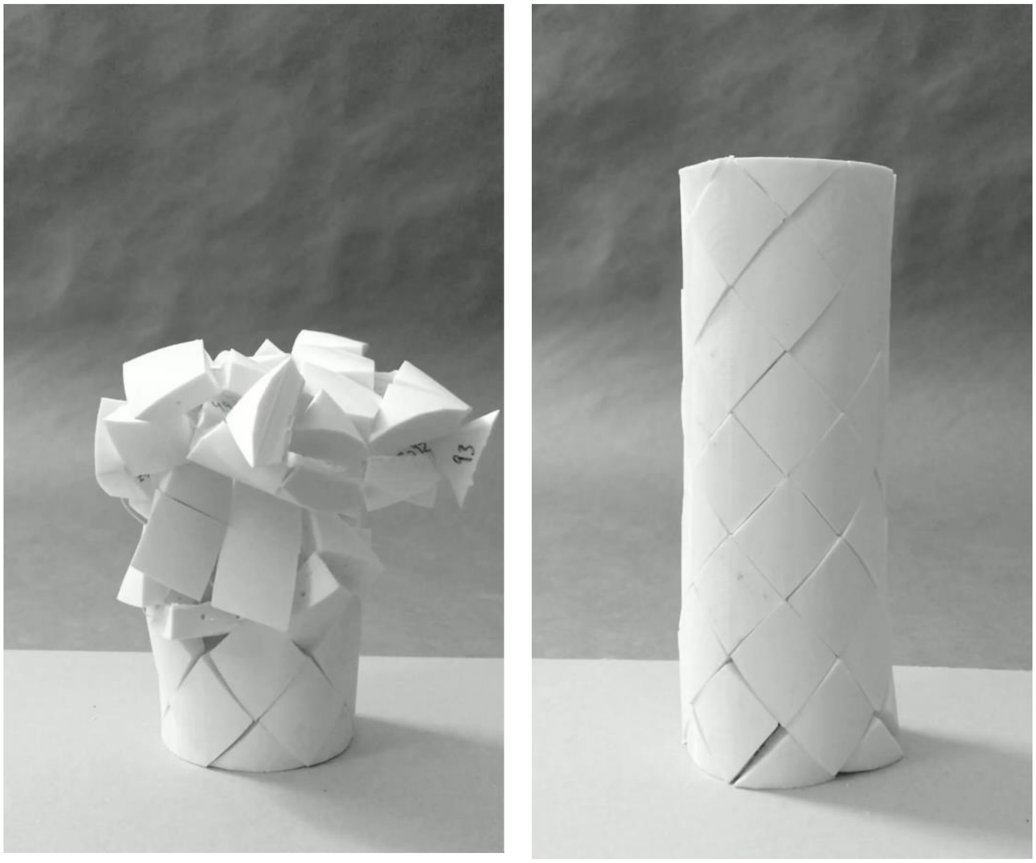


Figure 16. Cylindrical geodesic prototype exhibiting all X-type beads for greater geometric constraint. Self-repairing qualities are seen. (a) before, and (b) after tensioning.

Tortuosity

Minimizing the cumulative change in angle of a thread’s path presents another possible criterion, since static analyses reveal that sharp angles can hinder force transmission by more than 70% (Figure 13). Hence more obtuse angles will also minimize the force required to tension the bead system. In any network in which all nodes are of even degree, we can break this problem into numerous smaller problems: for each node, we need to construct a matching among the incident edges such that the sum of the angles between matched pairs is minimized. This sub-problem can be solved as a linear program, and if solved for each node independently the resulting global threading path is guaranteed to minimize sharp turns in the threading path. In the special case of a 2-cord bead system, like those discussed above, all nodes have even degree by default. Even degree frameworks also permit the use of the much simpler “Fleury’s Algorithm” for generating an Euler Tour, which is a single path that crosses each edge in a graph exactly once, necessarily in the shortest possible way. Fleury’s Algorithm therefore perfectly solves the Route Inspection problem for this special subcategory of frameworks.

These methods assume the existence of a static framework; however, we could also begin from a surface and place nodes and edges within this surface in a manner conducive to more optimal threading paths. For example, a geodesic naturally minimizes the cumulative change in angle of a path between two points on a surface, hence a geodesic grid shell, though it may require many independent cords, compensates with a comparatively low activation energy. The tubular models in Figure 16 offer one example of this approach, as helices are geodesics on a cylinder.

CHAPTER 6
SIMULATION

Dynamic architectural systems place unique demands upon the designer, to understand their behavior in time as a continuum and subject to a complex array of forces. Whereas physical scale models traditionally served this purpose, to mitigate rapid-prototyping turnaround times and quickly validate hypotheses it is helpful to examine an assembly in simulation first. For our purposes, the key actors are strings and solids, so we must be able to accurately simulate them both separately and in collision. String physics simulation methodologies generally fall into two categories: soft-body and rigid-body approximations. The latter approach treats a string as a chain of many small rigid bodies, each affixed to the next by a virtual joint. While soft-body approximations are specialized for deformable matter and may better simulate a string in isolation, simulations of beadwork are more reliable when strings and beads are approximated by the same type of object. Thus, we treated a string as a chain of many small rigid bodies, each affixed to the next by a point-to-point constraint, since this approach has been validated for speed and accuracy (de Jong et al. 2014) (Gołębiowski, et al. 2016).

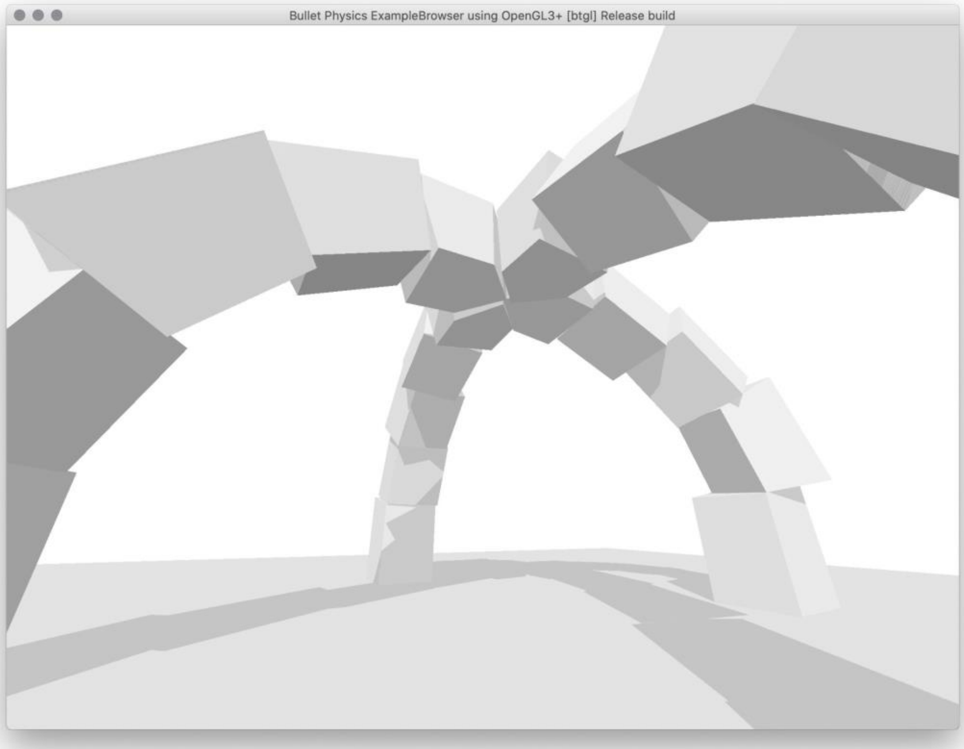

Bullet Physics ExampleBrowser using OpenGL3+ [btgl] Release build

Figure 17. Still from rigid-body simulation.

Pipeline

We developed a pipeline from Rhino3d and Grasshopper to the physics engine Bullet (Figure 17) (Coumans and Bai 2017). While the primary use of this engine is for game physics and machine learning for robotics, its robust rigid body collision physics makes it a natural choice for simulating string-bead interactions. Specifically, we used the engine's Python binding, "PyBullet," as it allowed us to contain all the logic of setting up and running a simulation in a simple body of text. This text is assembled dynamically in Grasshopper from referenced geometry and user-specified values and then exported to a .py file in a user-specified directory. The file can then be run from most code editors or the command line, like any Python script.

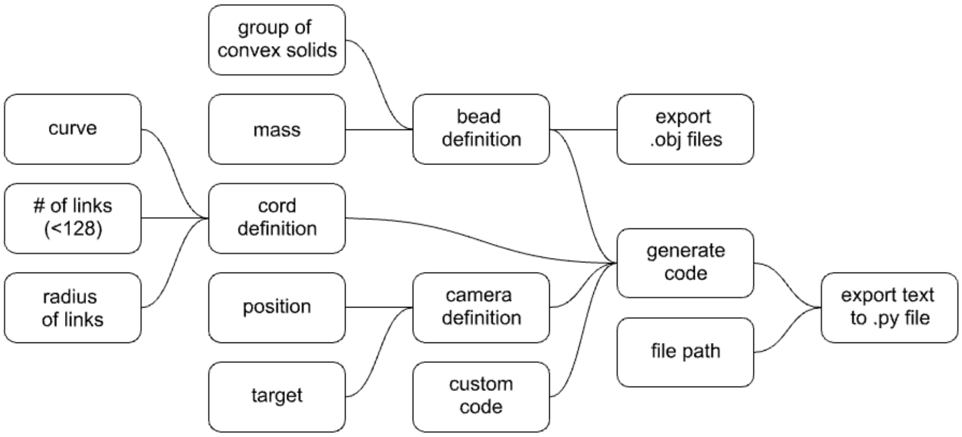


Figure 18. Rhino/Grasshopper to PyBullet simulation pipeline flow diagram.

Use Case

The use of this engine allowed rapid iteration through many bead geometries, helping to guide time and resources toward producing physical models that were more likely to yield interesting behavior. An example of a successful application of the simulation was in reproducing and expanding on a simple phenomenon we discovered in 1d sequences of mitered beads. We not only reproduced the observed behavior of 180-degree alternation, but also, we extrapolated this to create a 90-degree alternating sequence without having to physically print or otherwise manufacture a model for testing.

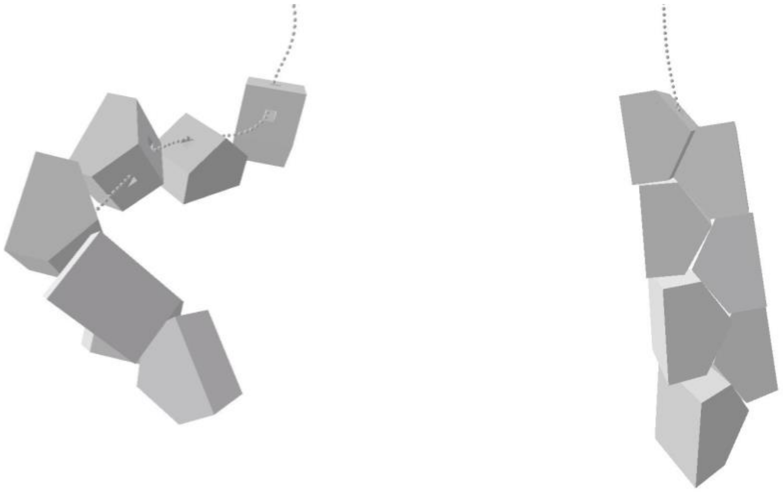


Figure 19. Simulation of bead shape that spontaneously arranges in an alternating pattern, (a) before, (b) after

Challenges

Unfortunately, the maximum number of links in a physics multibody is limited by the global variable `MAX_DEGREE_OF_FREEDOM`, which isn't exposed to the end user and instead is kept capped at 128 in the engine binding's current edition. This limit was found by trial and error and confirmed by the engine's original author Edward Coumans through its associated online forum.

One difficulty of simulating beads and strings, especially when realtime interactivity is desired, is the computational burden produced by simulating collisions between concave mesh objects such as hollow beads. A standard workaround is to decompose each concave mesh into a series of convex parts whose collisions can be calculated much more quickly and accurately. These parts are then constrained to move rigidly with one another so that together they behave like the original, concave object. Our simulation workflow employed both manual and automatic convex decomposition functionalities. Although we used custom scripts for higher accuracy, general algorithms such as VHACD are widely used (Mamou 2014). Realtime interactivity in the simulations enabled invaluable feedback between in silica experiment, human understanding, and physical prototypes.

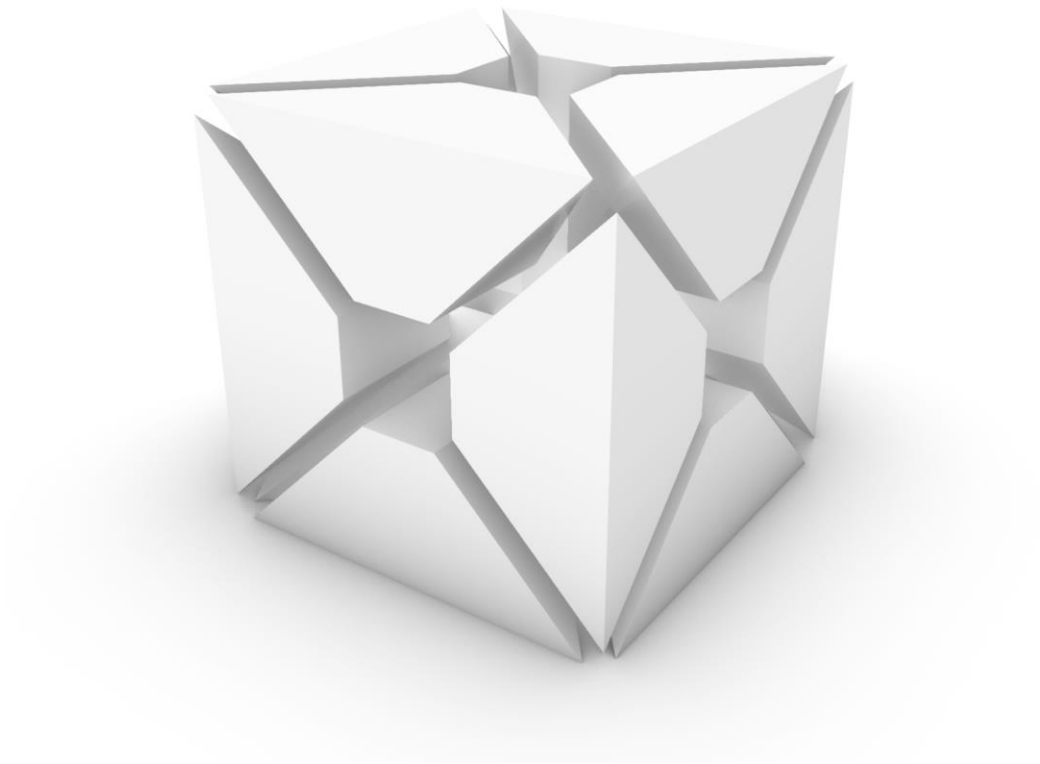


Figure 20. An example of convex decomposition of a cube-shaped bead with channels passing through it on all three axes.

The standard physics simulation solution in Grasshopper is Kangaroo (Piker 2017), however, this is a generic, multi-purpose dynamic relaxation solver, not a true rigid-body simulator. PyBullet presents some advantages by comparison. Screen picking, where a user clicks on objects and applies forces in the simulation by dragging across the screen, is an intuitive and highly interactive way to engage with a scene, which is less fundamental to Kangaroo, though in certain cases still achievable. The debugging options of PyBullet also offer the possibility of viewing bounding boxes and other information that may be hidden by a comparable Kangaroo simulation, which can help to identify the source of apparent errors. Also, most importantly, forces, the camera, and timing are programmatically controlled, so unlike in Kangaroo, events can be easily scheduled, sequenced, and animated by precise access to the time domain.

CHAPTER 7
SMALL-SCALE MODELS

1-Dimensional Sequences

1d sequences of beads, while simple topologically, help to clarify the geometric contributions of shape programming. For example, orientations need not be maximally constrained to exhibit self-organization. As mentioned in chapter 6, self-assembly of a regular 180˚ alternating pattern was observed in a simple 1d sequence of beads subject to agitation and gravity, in both physical tests and simulation (Figure 21). Gravity was key in this case, as it created a preference among the arbitrary possible orientations, where the center of mass of each bead tended to be as low as possible to minimize potential energy. We hypothesized we could use the same principle to create any angle of alternation, and with our simulation pipeline we were able to instantly validate this intuition.

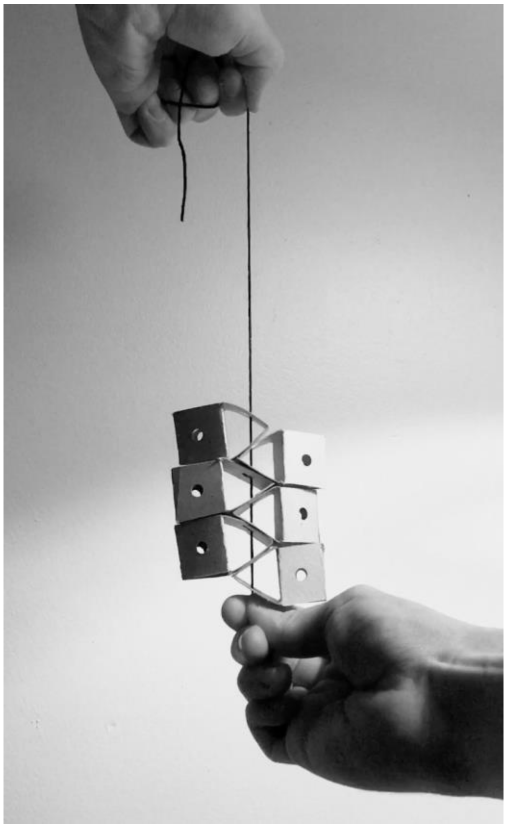


Figure 21. 1d 180-degree alternating sequence forms from random shaking under the influence of gravity.

We 3d printed sequences of beads with various kinds of interfaces, and discovered that screw-type interfaces were the most difficult to coax into their desired orientation, even though they technically fully constrain orientation. Lock-and-key type interfaces were adequate, but multi-cord interfaces could be just as effective if not better.

Another 1d sequence of beads demonstrated the power of potentially using multiple threadings, tensioning one or the other to produce different rigid outcomes from the same beads. One cord passed straight through each box-shaped bead, producing a single long rigid rod when pulled. The other passed through one of the beads at a right angle, producing a 90-degree turn in the rigidified sequence.

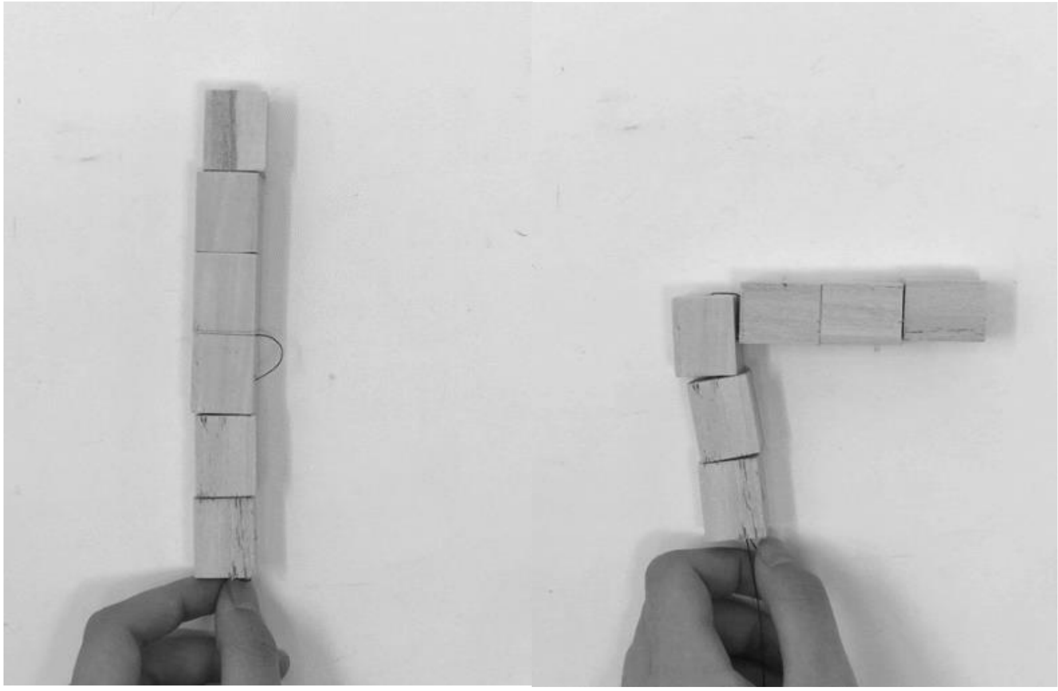


Figure 22. Multi-cord sequence with behavior dependent on which cord is tensioned.

A small indeterminate "arch" model helped demonstrate the effect of force minimization. Rather than a smooth arc, it formed a limited number of "kinks" in the sequence of beads. Repeatedly slackening and tensioning this arch, we observed that while the arch shape was different each time, certain arrangements of kinks were more likely to form in proportion with how much they minimized tension, or equivalently how much they maximized the angles at the joints. This probabilistic model strikes a balance between indeterminacy and order. Conversely, for a set of *n* kinks at specified positions in the sequence of beads, the angles formed at the respective kinks are consistently the same. We hypothesized that these angles are such that, if we view the arch as an *n*+1 -bar linkage in the plane, then the sum of the squares of the angles between adjacent members is maximized.

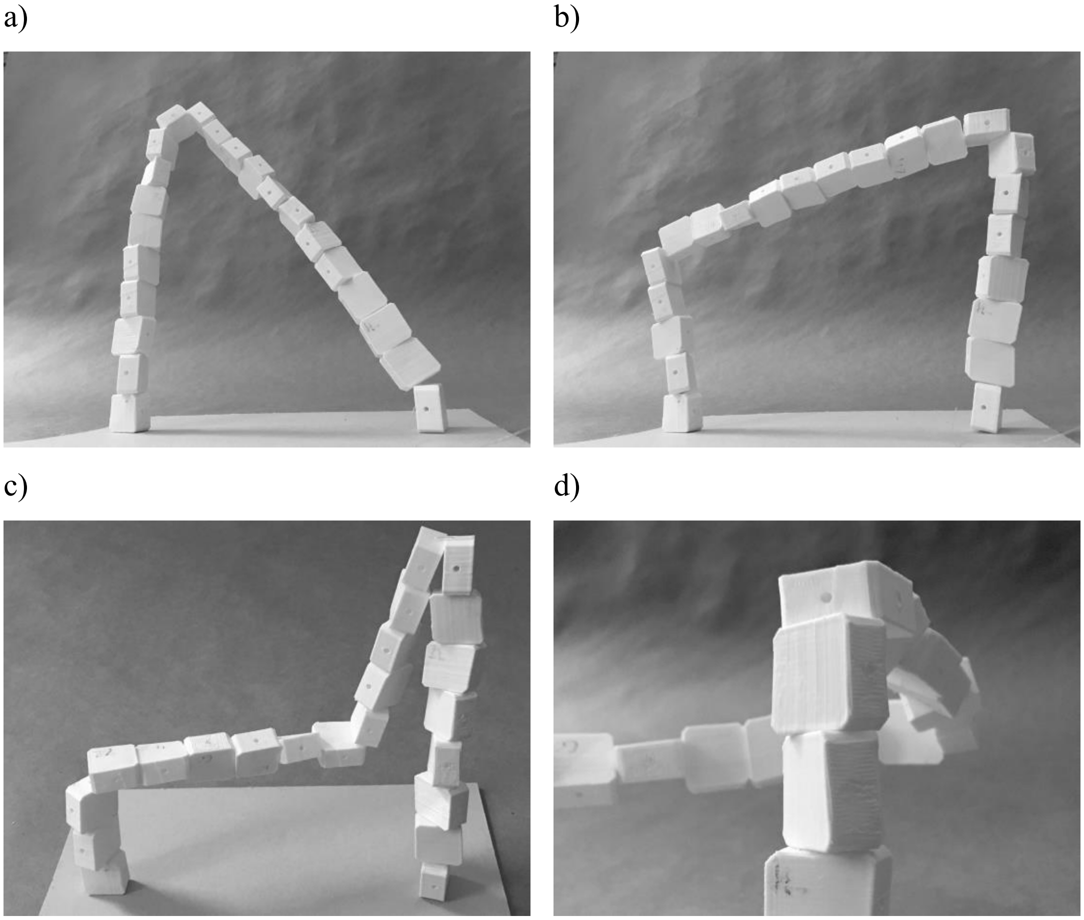


Figure 23. A simple, indeterminate arch exhibiting multiple stable states (a-c) which appear more or less frequently in accordance with how they minimize a function of the kink angles. (d) shows a closeup of a kink.

2.5-Dimensional Shells

As an early test we applied our techniques to a simple dome-shaped grid shell structure. The beads were 3d printed in generic polylactic acid (PLA) thermoplastic, and the threading was performed by hand with cotton cord. The first prototype had a footprint of approximately 5 in × 5 in. When the cords at each of the four feet were pulled, the beads spontaneously formed a rigid dome structure (Figure 24). This success motivated a second prototype at approximately three times scale, with a 15 in × 15 in footprint, this time threaded such that two strings passed through each bead. This gave more control at the expense of increased friction. To compensate, the dome's assembly was minimally aided (Figure 25). When tension was removed from the upright dome, the slightest touch would collapse the structure, demonstrating that this is not a true funicular dome but a tensegrity reliant on tension and compression in balance. Conversely, starting from a slack state, as the dome's apex was lifted into place simultaneously with a gradual reintroduction of tension, the remainder of the structure assembled of its own accord. This "one-handed" assembly behavior not only demonstrates the influence of threading path and friction on global behavior but also promises further use cases. For example, as a counterweighted window similarly retains its position after it is adjusted, a heavy structure could form and unform with only gentle assistance. This could potentially lead to a novel kind of masonry construction with improved safety, user-friendliness, and sustainability.

We also investigated shell structures wherein the beads formed frames for planar panels, as opposed to as nodes with radiating spokes. These behave as a kind of graph theoretic dual to the frameworks considered thus far. The model shown in Figure 26 was additionally attached to a fabric skin, and the tension was applied as a lacing pattern around the perimeter of the dome.

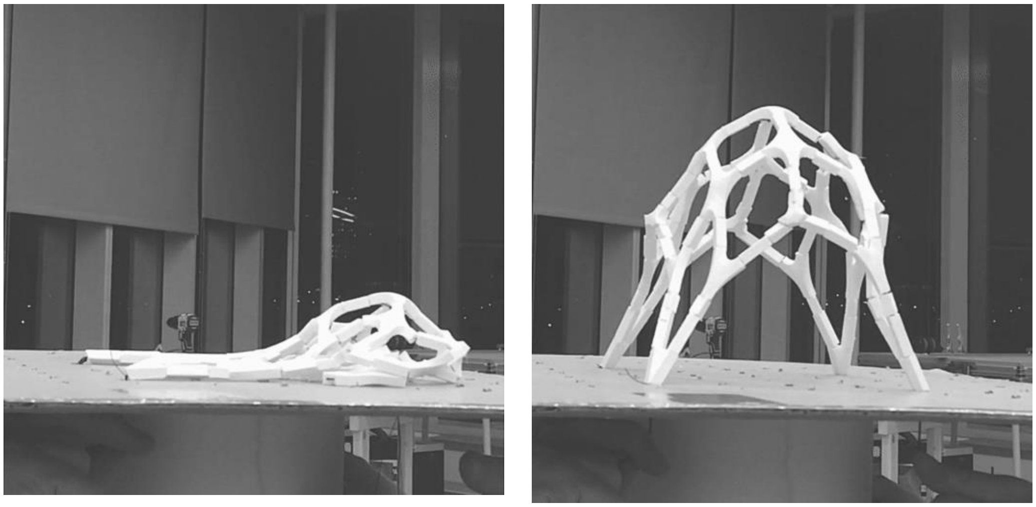


Figure 24. Initial dome prototype, (left) before, and (right) after tensioning cord

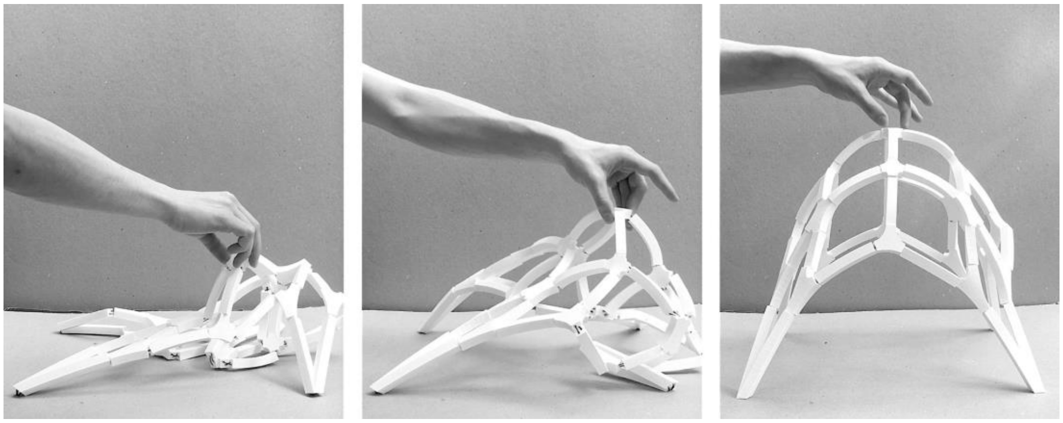
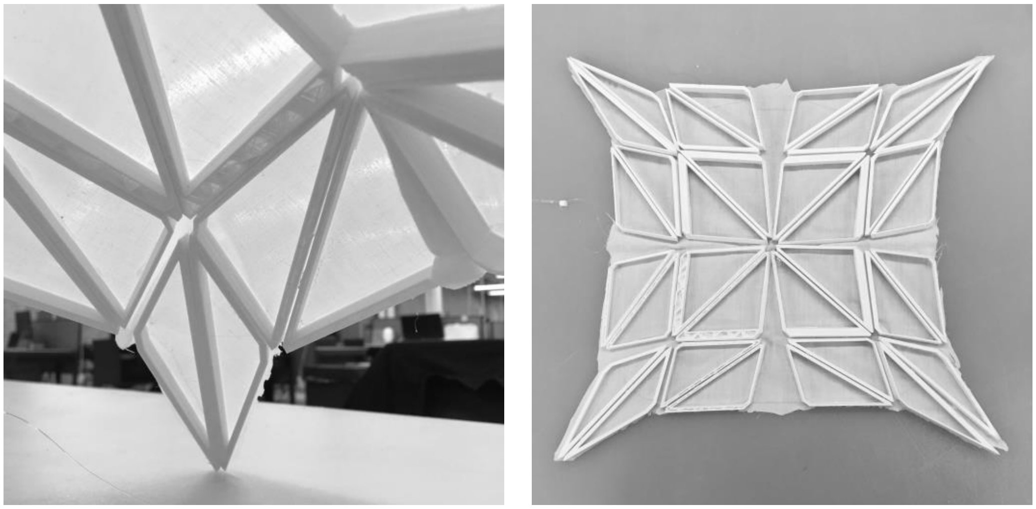


Figure 25. 15" counterweighted 2-cord model. A beadwork assembly can order itself into a self-standing structure with little to no aid.



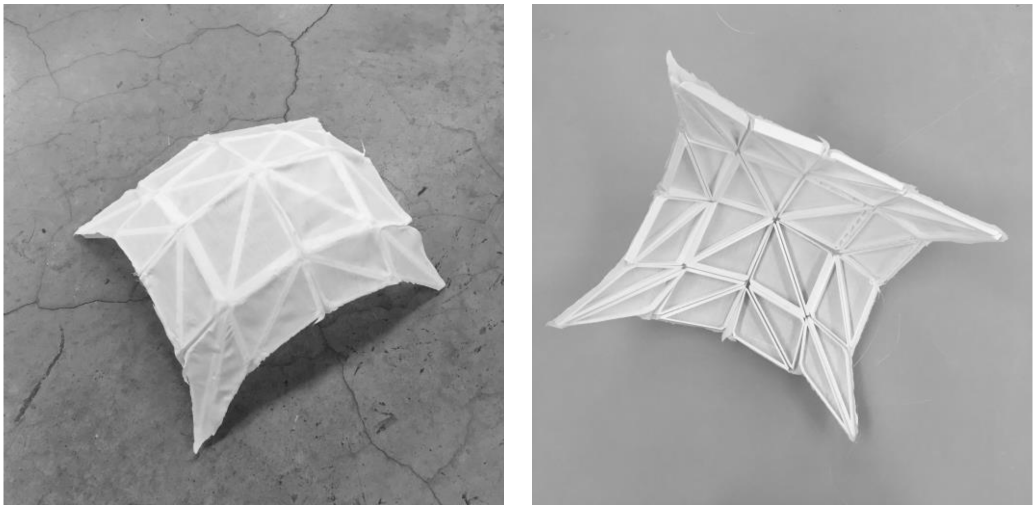


Figure 26. Triangulated shell model, composed of rigid triangular frames. Tensioned by nylon cord sewn through attached fabric along the periphery.

CHAPTER 8
LARGE-SCALE FABRICATION

Tool Design

3d printed beads are limited by the size of the printer bed, and more importantly they would take an extremely long time to produce at architectural scale. As an alternative, expanded polystyrene (EPS) foam, sometimes used interchangeably with the proprietary name “Styrofoam™,” is a good candidate for large-scale beads because of its high stiffness-to-weight ratio. Aside from physical cutting and milling, one of the most common methods of carving EPS is hot wire cutting. These cutters function by running an electrical current through special resistance wire such as Nichrome or René 41, causing it to heat up to and beyond the melting point of foam. This wire can be mounted on various kinds of tools, from handheld frames to tabletop jigs to CNC machines. Capitalizing on the precision of CNC tools, we elected to utilize the AAP Rand Fabrication Facilities’ ABB IRB 4600 6-axis robotic arm as a mount for our hot wire tool. Numerous precedents exist for this kind of pairing of robot and tool: The University of Michigan TCAUP FABLab’s Kuka 6-axis robot was fitted with a hot wire cutter on a U-shaped frame of double and quadruple T-slotted aluminum beams for the fabrication of the Periscope Tower by Matter Design Studio. A similar T-slotted aluminum frame was attached to the ABB IRB 2400-16 at the Design Ecologies Laboratory at Penn State University for research in robotic stereotomy ([2]). The hot wire cutter tool designed by Formeta at Ball State University used 1” steel pipe shaped instead of aluminum and made novel use of guitar string tuning pegs as an adjustable tightening mechanism for the wire.

[2] http://papers.cumincad.org/data/works/att/caadria2015_226.content.pdf

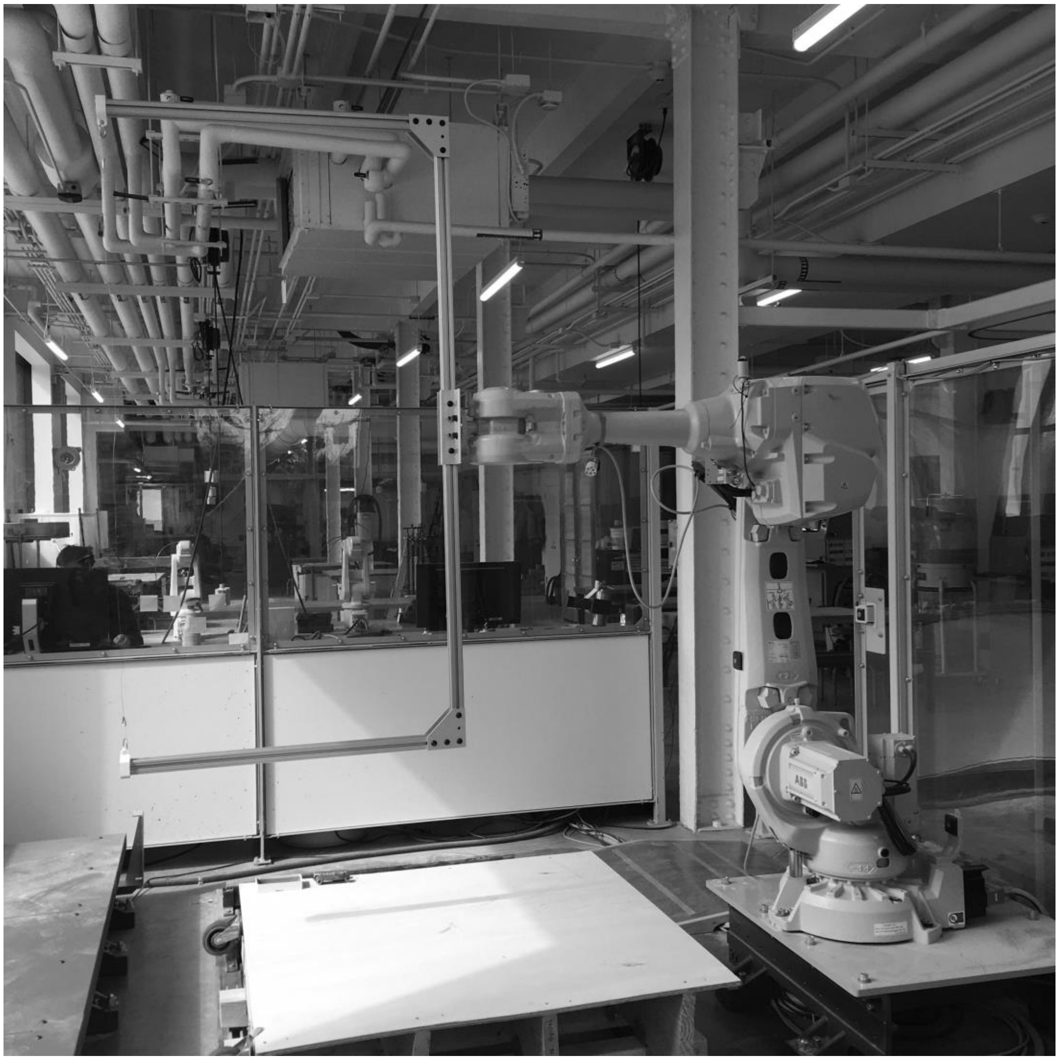


Figure 27. ABB IRB 4600 6-axis robotic arm at Cornell AAP Rand Hall fabrication lab, with custom 6' hot wire cutter end effector mounted to flange.

Due to its modularity and strong precedents, we also elected to construct the armature of our tool from T-slotted aluminum. We used 1.5" square cross-sections to construct a U-shape 6' wide and 3' deep, using 90-degree angle plates to rigidify the connections and preserve tension in the wire. The major dimensions of the tool were partly influenced by the size of the largest foreseeably acquirable EPS foam blocks, which at 2' x 4' x 8' would need to fit within its span. For the wire itself, we selected 24g Nichrome with power supplied by a 36V transformer, capable of heating the wire to more than 600 degrees Fahrenheit. We also selected a common light switch dimmer as a means of controlling the current in the wire, and thereby the temperature, dynamically. This would prove useful to reduce unwanted slack in the wire mid-cut by slightly lowering the temperature, or to make the wire cut more smoothly or quickly by raising the temperature.

We 3d printed two custom end caps able to slip onto the open ends of the T-slotted aluminum armature. The wire was attached to each of these at its two ends, held in place by small brass ferrules of the kind used with certain stringed musical instruments. Compression springs were used as part of the assembly of the endcaps to ensure the uniform tautness of the wire before and after applying current by taking up some of the slack that appears as the Nichrome expands with heat. The design is slip-on and held securely in place by friction and a load mostly perpendicular to the direction of sliding. The PLA plastic is a good insulator, helping to reduce the risk of charge leaping from the wire to the armature itself and shorting the circuit. These end caps were additionally designed to support accessories, such as the vacuum hose mount that forms an adaptive friction fit with most standard vacuum nozzles and can be swiveled so as to capture as much toxic fumes as possible from the melting polystyrene. Polystyrene releases styrene gas when melted, a known carcinogen, requiring all cutting to be done in a well-ventilated area. The placement of the vacuum hose at the end cap was based on the observation that while cutting a piece of polystyrene, most smoke and fumes escape from the points where the wire is entering and exiting the block.

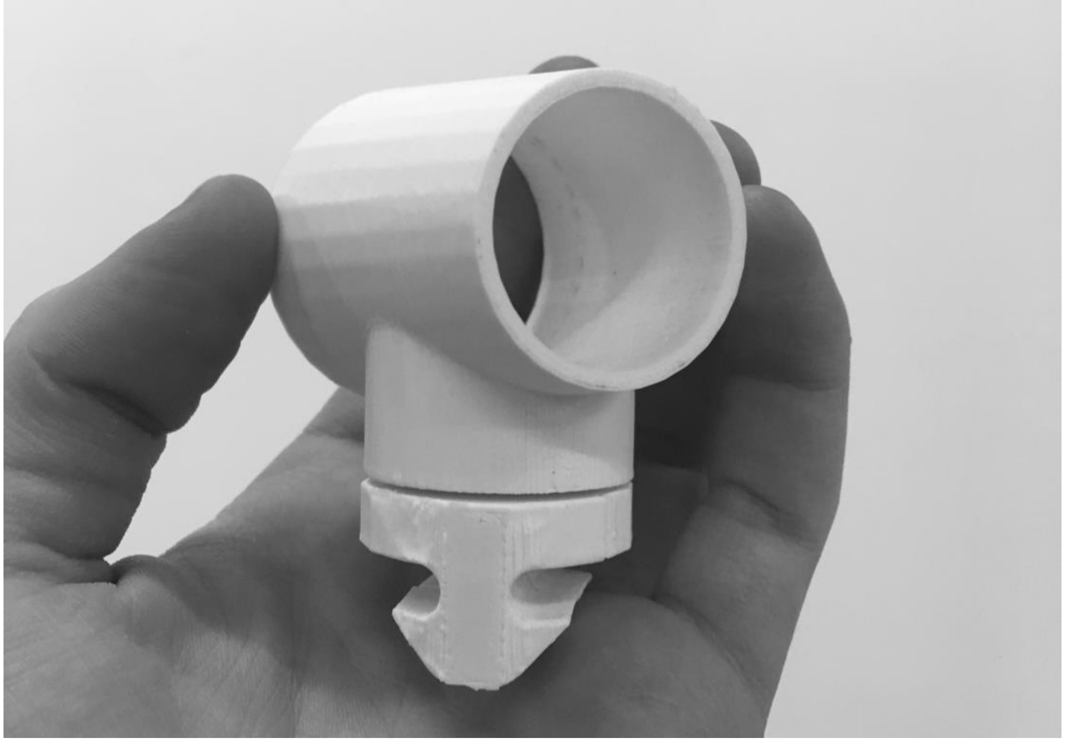


Figure 28. Swiveling friction-fitted vacuum hose attachment for fume collection, 3d printed in PLA plastic.

Fabrication Strategy

With these tools, new strategies were needed for carving bead geometry. One primary constraint was the need for all surfaces of the bead to be *ruled*, that is, parameterizable as a surface swept out by a line in space. Beads could either be designed this way deliberately from the outset, or else they could be approximated with a ruled surface after the fact. In the context of our graph-augmented beads this is easier than the general case. “Edge” beads with just a single channel through them or several in parallel are created by sweeping cross section polygons, so that each face is the surface swept out by one of the polygon’s edges. “Node” beads, with more complex connectivity, use the “sweep” command for most faces, but their top and bottom surfaces are often not developable. If these surfaces are approximately flat, we can take the intersections of the surface boundary with a set of planes perpendicular to the surface arrayed linearly along their normals. These pairs of points can be connected with lines which can then be lofted with linear interpolation to produce a piecewise ruled surface close to the shape of the original.

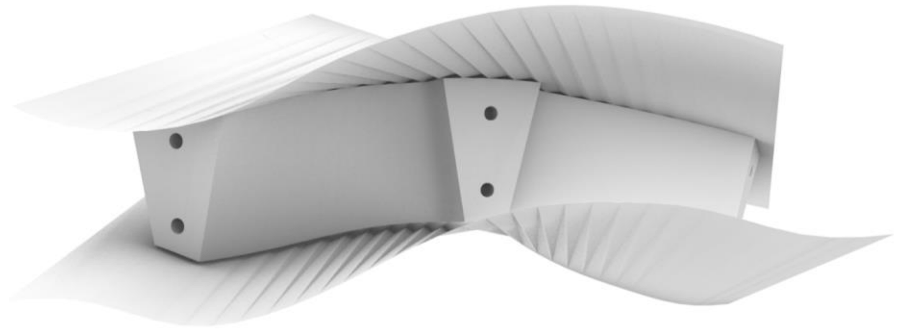


Figure 29. Two ruled surfaces approximating the doubly-curved patch surfaces of a standard node bead.

Internal channels would also have to be carved from ruled surfaces, so a true piped curve, that is, a channel carved by sweeping a circle through the path of the cord, would be unachievable with hot wire cutting. Instead, we proposed two possible approaches. The first strategy was to decompose a bead by slicing with ruled surfaces that pass through its intended channel cuts, then carve out the two halves of the channel on either of the resulting pieces before adhering the pieces together again, either with glue or by lashing them with cords around the outside. This process of slicing and recombining recalls the "convex decomposition" process used to pre-process digital bead geometries for rigid body simulation, in a natural extension of the simulation-to-physical prototyping workflow. While this could achieve more elegant geometry for the channels, with node beads of high degree the decomposition could quickly become quite complex, and the structural integrity of the bead may be jeopardized by such fragmentation.

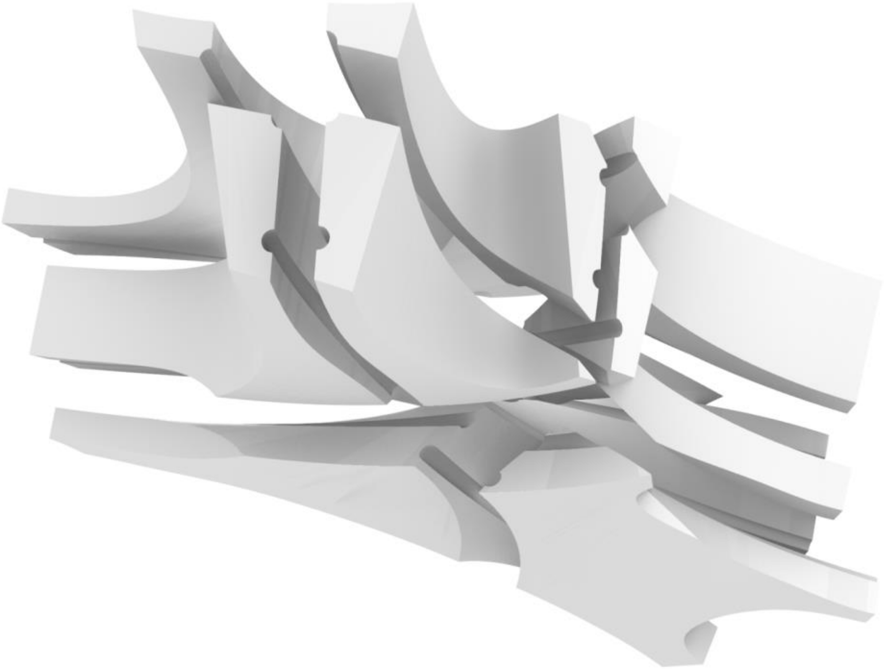


Figure 30. A bead with 5 independent internal channels, decomposed with ruled surfaces, shown exploded.

The second method proposed was to simply cut straight cylindrical channels. This would constrain the outer bead geometry somewhat, as for instance one cannot carve a straight path around a concave corner. However, this also meant that the channel could be carved without completely breaking the bead into two or more separate pieces, by inserting the wire along a plane, carving the channel, then removing the wire along the same plane. While this would create a narrow gap through which the cord might escape its channel, it could be closed off with adhesive, or a tubular reinforcement could be inserted into the entire channel or just its ends to prevent such slippage. For example, in our final full-scale model we used ¾” PVC pipe segments as a reinforcement, adhered to the inside of either end of each channel.

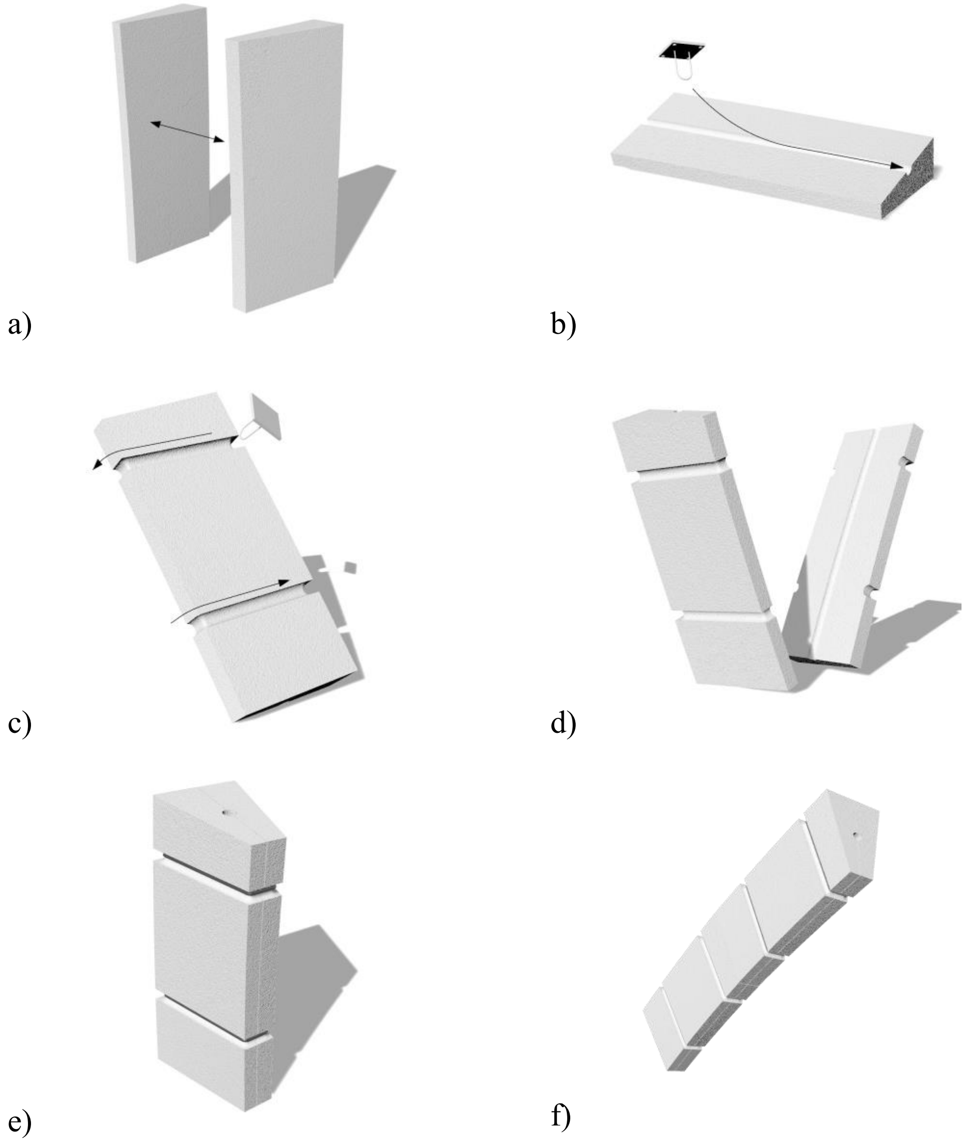


Figure 31. Left-to-right, top-to-bottom: (a) A bead is divided by a ruled surface bisecting the channel. (b) A rigid hot wire loop is used to carve each half of the channel, from either of the resulting halves of the bead. (c) The same tool is used to care shallow channels on the outside of the bead, for guiding the lashing cords.

A visualization of what a large format bead test structure might look like is shown in Figure 32. Such a structure could provide rigid shelter while using no mortar and yet still be safe and robust to unusual loads such as high winds or children climbing upon it.

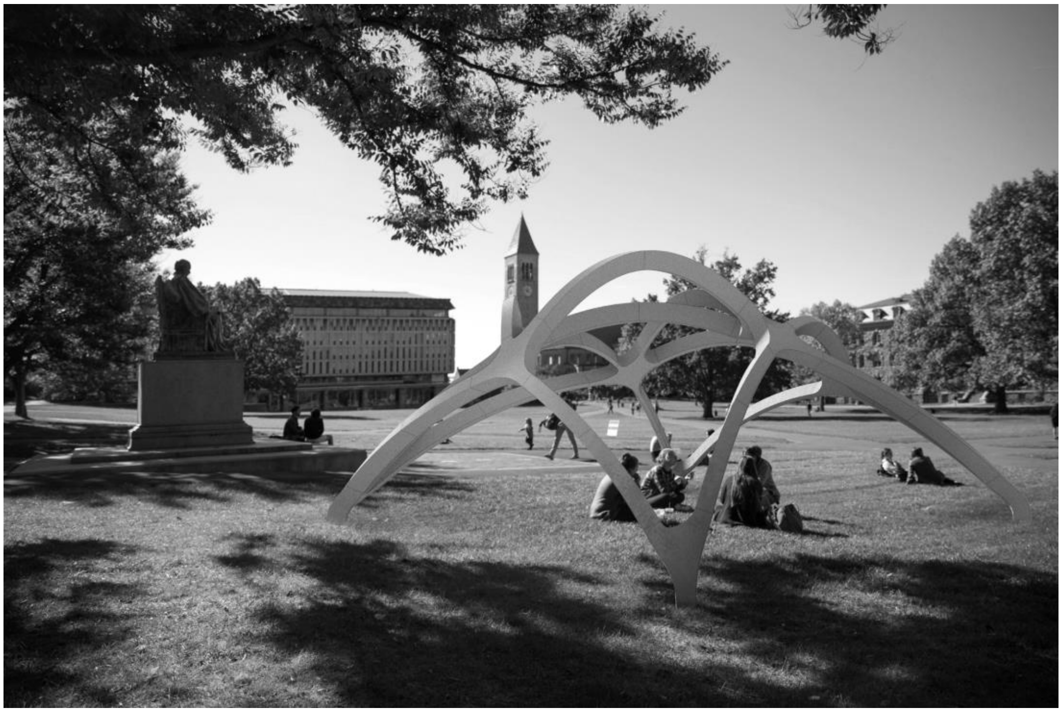


Figure 32. Speculation of what a large-format bead structure might look like.

This structure uses unique geometries for every bead, though, which greatly increases fabrication time. Taking advantage of symmetry, we proposed a simpler ring-shaped geometry for our first full-scale test, since we would only need to design the toolpaths for cutting one bead and then run those routines repeatedly on the robot. This ring was 2.9m in diameter and formed of 20 beads, each of which was as large as could be fit in the raw volume of the foam blocks available, which were each about 18” x 10” x 10”. Figure 31.a. shows the completed prototype. Notice that there is some unaccounted-for slack, manifesting as a slight sag in the ring. We interpret this as having resulted from the cord cutting into the foam where it was tautened. Nevertheless, the structure had good integrity and even was able to roll over rough terrain. This example of mobility demonstrates a possible advantage of compliance and slack. Perhaps in the future a bead structure could be applied to specialized wheel devices for off-road vehicles or research rovers.

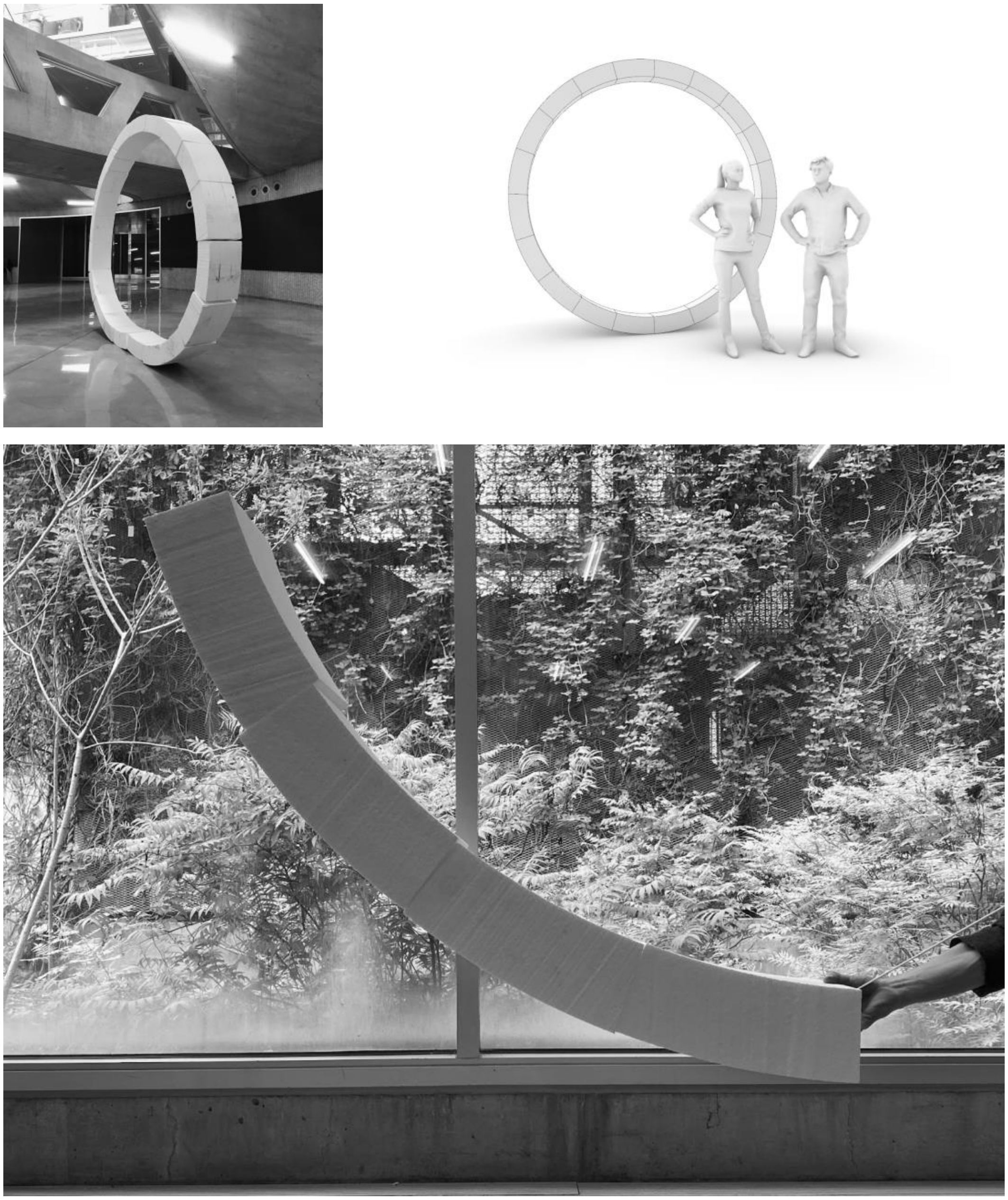


Figure 31. a) completed prototype, b) rendering with scale figures, c) 4-bead segment showing rigidifying properties, without mortar or other adhesive

CONCLUSION

In this preliminary study we examined the potential of beads as a medium for controlled self-assembly and deployable structures. The potential applications of this technique are diverse, including but not limited to:

- rapidly deployable structures for disaster relief, temporary installations, and leisure
- low-energy shape actuation for soft robotics and medical prosthetics, such as the CardioARM medical snake robot that uses a similar mechanism to control the joints of a many-segmented probe (Ota et al. 2009)
- assembly from-a-distance for autonomous structures in extreme environments as suggested by projects like MIT Self-Assembly Lab's "Aerial Assembly" project (Staback et al. 2017)

Rigid body simulation has also proven a valuable tool for rapid prototyping and experimentation. The small-scale 2.5d prototypes presented in this paper demonstrate some potential challenges of threading and scale that must be further investigated in order to validate these concepts for human-scale applications. Whereas graph augmentation offers a generic and versatile framework for design, further research should leverage our current understanding of forces and complexity to produce more robust specimens. Our approach to self-assembly is also virtually agnostic to the materials used, such that even off-the-shelf materials are sufficient to demonstrate the behaviors discussed, but in scaling up the combination of stiffness and lightness will place more constraints on the variety of eligible materials. In the context of existing research on self-assembly for architecture, though, this is a departure from the specialized handshake mechanisms heretofore seen, such as magnets, bimetals, shape-memory polymers, and thermo- or hygro-active materials. The self-assembly of beads offers a novel approach with potentially greater control of shape formation, tunable and reversible rigidity, and structural robustness.

REFERENCES

Bier, Henriette. 2011. "Robotic Environments." In Proceedings of the 28th International Symposium for Automation and Robotics in Construction (ISARC), Seoul, South Korea, 29 June – 2 July 2011, 863–868. IAARC.

Clifford, Brandon, and Wes McGee. 2011. "Periscope Foam Tower." In Fabricate 2011, edited by Ruairi Glynn and Bob Sheil, 77–80. London: UCL Press.

Coumans, Erwin, and Yunfei Bai. 2018. Pybullet. https://pybullet.org/wordpress. V.2.88. PC.

Dear, Tony, Blake Buchanan, Rodrigo Abrajan-Guerrero, Scott David Kelly, Matthew Travers, and Howie Choset. 2020. "Locomotion of a Multi-Link Non-Holonomic Snake Robot with Passive Joints." The International Journal of Robotics Research 39 (5): 598–616.

de Jong, J., K. Wormnes, and P. Tiso. 2014. "Simulating Rigid-Bodies, Strings and Nets for Engineering Applications Using Gaming Industry Physics Simulators." Paper presented at i-SAIRAS: International Symposium on Artificial Intelligence, Robotics and Automation in Space, Montreal, June 17-19. European Space Agency.

Edmonds, Jack, and Ellis L. Johnson. 1973. "Matching, Euler Tours And the Chinese Postman." Mathematical Programming 5 (1): 88–124. https://doi.org/10.1007/bf01580113.

Gołębiowski, W., R. Michalczyk, M. Dyrek, U. Battista, and K. Wormnes. 2016. "Validated Simulator for Space Debris Removal with Nets and Other Flexible Tethers Applications." Acta Astronautica 129: 229–240. https://doi.org/10.1016/j.actaastro.2016.08.037.

http://de.wikipedia.org/wiki/Benutzer:Drahkrub, 2010. *Push puppet giraffe. Left: strings strained (plate internally fixated - not visible). Right: strings released*. [image] Available at: https://upload.wikimedia.org/wikipedia/commons/1/1e/Push_puppet_giraffe_%28dkrb%29.jpg [Accessed 2 August 2021].

Mamou, Khaled. 204. V-HACD: Volumetric-Hierarchical Approximate Convex Decomposition. V.2.0. PC.

Mitchell, Stuart, and Iain Dunning. 2020. PuLP: A Linear Programming Toolkit for Python. V.2.0. PC.

Ota, Takeyoshi, Amir Degani, David Schwartzman, Brett Zubiate, Jeremy McGarvey, Howie Choset, and Marco A. Zenati. 2009. “A Highly Articulated Robotic Surgical System For Minimally Invasive Surgery.” The Annals of Thoracic Surgery 87 (4): 1253–1256. https://doi.org/10.1016/j.athoracsur.2008.10.026.

Piker, Daniel. *Kangaroo Physics*. V.2.42. https://www.food4rhino.com/en/app/kangaroo-physics. PC. 2017.

Roth, B., and W. Whiteley. 1981. "Tensegrity Frameworks". *Transactions Of The American Mathematical Society* 265 (2): 419-419. doi:10.1090/s0002-9947 1981-0610958-6.

Rutten, David. Grasshopper. V.1.0.0007. Robert McNeel & Associates. PC. 2020.

"Self-Folding Proteins — Self-Assembly Lab". 2021. *Self-Assembly Lab*. https://selfassemblylab.mit.edu/proteins/.

Senatore, Gennaro, Philippe Duffour, Pete Winslow, and Chris Wise. 2017. “Shape Control and Whole-Life Energy Assessment of an ‘Infinitely Stiff’ Prototype Adaptive Structure.” Smart Materials and Structures 27 (1): 015022. https://doi.org/10.1088/1361-665x/aa8cb8.

Sobek, Werner, and Patrick Teuffel. 2001. "Adaptive systems in architecture and structural engineering." In Smart Structures and Materials 2001: Smart Systems for Bridges, Structures, and Highways, vol. 4330, pp. 36-45. International Society for Optics and Photonics, 2001.

Staback, Danniely, MyDung Nguyen, James Addison, Zachary Angles, Zain Karsan, and Skylar Tibbits. 2017. “Aerial Pop-Up Structures.” In ACADIA 2017: Disciplines & Disruption [Proceedings of the 37th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA) ISBN 978-0-692-96506-1], Cambridge, MA 2-4 November, 2017, 582–589. CUMINCAD.

Tibbits, Skylar. 2011. “A Model for Intelligence of Large-Scale Self-Assembly.” In ACADIA 11: Integration through Computation [Proceedings of the 31st Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA) ISBN 978-1-6136-4595-6], Banff (Alberta) 13-16 October, 2011, 342-349. CUMINCAD.

Tibbits, Skylar. 2014. “Fluid Crystallization: Hierarchical Self-Organization.” In Fabricate 2014, edited by F. Gramazio, M. Kohler, and S. Langenberg, 297 303. Zurich: UCL Press