

Computational Design of Telescoping Structures

CHRISTOPHER YU, Carnegie Mellon University

KEENAN CRANE, Carnegie Mellon University

STELIAN COROS, Carnegie Mellon University

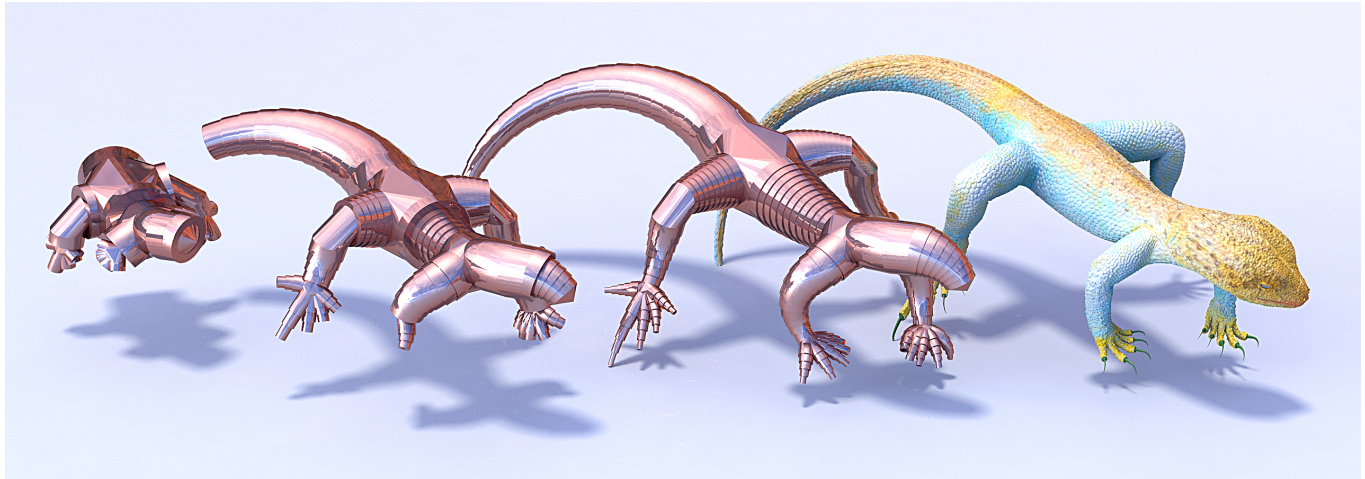


Fig. 1. A telescoping lizard in various stages of extension approximates an input surface (right). We parameterize telescoping structures as networks of smooth space curves with special geometric properties, allowing users to rapidly explore the space of telescoping designs.

Telescoping structures are valuable for a variety of applications where mechanisms must be compact in size and yet easily deployed. So far, however, there has been no systematic study of the types of shapes that can be modeled by telescoping structures, nor practical tools for telescopic design. We present a novel geometric characterization of telescoping curves, and explore how free-form surfaces can be approximated by networks of such curves. In particular we consider piecewise helical space curves with torsional impulses, which significantly generalize the linear telescopes found in typical engineering designs. Based on this principle we develop a system for computational design and fabrication which allows users to explore the space of telescoping structures; inputs to our system include user sketches or arbitrary meshes, which are then converted to a curve skeleton. We prototype applications in animation, fabrication, and robotics, using our system to design a variety of both simulated and fabricated examples.

CCS Concepts: • **Computing methodologies** → **Parametric curve and surface models**; • **Applied computing** → Computer-aided design;

Additional Key Words and Phrases: telescoping structures, deployable structures, computational design, fabrication, discrete differential geometry

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE1252522. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM. 0730-0301/2017/7-ART83 \$15.00
DOI: <http://dx.doi.org/10.1145/3072959.3073673>

ACM Reference format:

Christopher Yu, Keenan Crane, and Stelian Coros. 2017. Computational Design of Telescoping Structures. *ACM Trans. Graph.* 36, 4, Article 83 (July 2017), 9 pages.

DOI: <http://dx.doi.org/10.1145/3072959.3073673>

1 INTRODUCTION

A pirate's telescope, consisting of straight, nested cylinders, is a familiar sight commonly associated with tales of seafarers and explorers. The simple telescoping mechanism behind these so-called *spyglasses* has endured over the centuries, owing to its simple effectiveness for compact storage and rapid deployment, and is still widely used in modern engineering (Garrette and Ryan 1969; McCord and Williford 1966). Deployable structures have more broadly become important in applications where an unwieldy object must be stored in or transported through a smaller vessel, *e.g.*, large solar panels carried by space-bound vessels (Stinson 2014), or arterial stents that must travel through narrow passages during surgery (Kuribayashi et al. 2006). Generalized telescoping mechanisms likewise hold great promise for deployable design, providing a fundamentally new kind of joint that can be reshaped in surprising and entertaining ways. The design space of telescoping structures, however, remains relatively unexplored. This paper is a first foray into mathematical and computational models for generalized telescopes and their applications.

At the most basic level, a *telescoping structure* consists of a sequence of nested units that can be extended and retracted. Most modern instances of such structures consist of a linear sequence of



identical and parallel cylindrical shells. Traditionally, the linearity of telescopes may have arisen from optical considerations (as with the spyglass), but for purely mechanical applications, telescoping structures are capable of achieving a much broader and more varied range of shapes and motions. However, designing more complex telescoping mechanisms by hand would require considerable effort: factors such as shell shapes, dimensions, and orientations, that affect compactness and mechanical feasibility have to be taken into account for each of potentially numerous shells, making it difficult to iterate on design and explore the large space of possibilities.

In this paper, we therefore take a computational approach to telescoping design. The starting point is a novel geometric parameterization of telescoping structures—the key insights are that (i) infinitesimally, a telescoping motion must be a *screw motion*, meaning that each shell must follow a path of constant curvature and constant torsion (*i.e.*, a helix), and (ii) in a fully extended configuration, telescopes have additional flexibility that can be modeled as finite impulses of torsion. The problem of designing a telescope based on an arbitrary space curve then becomes the problem of approximating the curve by a G^1 piecewise helical curve. We formulate an optimization problem to compute such curves, which can then be segmented into telescoping shells.

We also present schemes for combining a network of such curves into larger telescoping structures. Using mesh skeletonization, we are able to semi-automatically produce telescoping approximations of a given input surface mesh, especially effective for tree-like surfaces. Finally, we develop a design tool that enables novice users to create telescoping structures, and use this tool to create a wide variety of examples, showcasing the flexibility of our parameterization to applications in animation, fabrication, and robotics.

2 RELATED WORK

Deployable structures. While the design of telescoping structures has not received much attention thus far, *deployable structures* in general are an active area of research in mechanical and civil engineering (Pellegrino 2002; Puig et al. 2010). While spaceflight has remained a primary motivator for such research, developments have also occurred in applications such as deployable robots (Salemi et al. 2006) and bridges (Rhode-Barbarigos et al. 2012), among other things. Recent work on the design side has also focused on ensuring that the motion undergone by a deployable or reconfigurable structure is physically feasible (Garg et al. 2016; Zheng et al. 2016).

Compact storage. A major motivation for telescoping structures is to enable objects to be collapsed into a compact form for storage or transportation. This also motivated earlier work by Li et al. on stackabilization (2012) and foldabilization (2015). Both approaches focus on modifying object geometry to enable a desired compact configuration – of a stack of objects in the former, and of a single folded object in the latter. These build upon previous work on design systems for 3D printable furniture (Saul et al. 2011; Umetani et al. 2012), where collapsibility and portability are often desirable traits.

Computational folding. Similar problems have also been studied in the area of computational origami and folding. One widely studied problem is to compute an origami folding pattern that produces a papercraft facsimile of an input 3D model (Tachi 2010). There have

also been recent advances in origami deployable structures (Cheung et al. 2014), as well as design of origami structures that are able to fold themselves, via exposure to heat (An et al. 2014) or microwaves (Yasu and Inami 2012).

Geometry of space curves. Space curves are a natural choice for modeling rod- or strand-like phenomena (Bergou et al. 2008). In this work, we model telescoping structures using piecewise helical curves with G^1 continuity; these so-called *super-helices* and their generalizations (Bertails-Descoubes 2012; Casati and Bertails-Descoubes 2013) have previously been used for hair simulation (Bertails et al. 2006). Although the problem of fitting super-helices to arbitrary curves has been studied before (Derouet-Jourdan et al. 2013), we are far more constrained in our ability to vary curvature along a path, motivating our need for a different method. In contrast, in the limit as we add more shells, our designs approximate space curves of constant curvature but arbitrary *torsion*. Beyond circular arcs and helices, special curves of constant curvature include *Salkowski curves* which have continuous (rather than constant) torsion (Monterde 2009); importantly, a theorem of Ghomi (2007) implies that any C^∞ curve is well-approximated by a curve of constant curvature, which helps to justify their use here.

Mesh skeletonization. As a starting point for approximation of general surface meshes, we use a 1D mesh “skeleton” which can be computed by approximating an analytic description (Dey and Sun 2006), or by iterative mesh contraction (Au et al. 2008; Tagliasacchi et al. 2012). Mesh skeletons are used as a 1D proxy for the surface in applications such as automatic segmentation and skinning (Au et al. 2008), or fitting of axially-oriented elements such as beads (Raab et al. 2004) or generalized cylinders (Zhou et al. 2015).

3 THE SPACE OF TELESCOPES

What shapes can a telescope have? Although our telescopes will ultimately be constructed from discrete collections of shells, we will consider this question from dual discrete and continuous viewpoints. This duality allows us to leverage differential geometry to understand the space of possibilities, and leads to efficient optimization problems that can incorporate free-form geometry.

3.1 Geometry of telescoping shells

At a high level, a *telescoping structure* can be thought of as a sequence of extensible components, referred to as *telescoping shells*. Intuitively, each shell represents a rigid piece of material with an interior cavity shaped to accommodate subsequent, smaller shells which we call *children*. Our definitions are guided by three common-sense axioms:

- (A1) Telescoping shells are rigid.
- (A2) Each shell has a continuous, collision-free motion between distinct retracted and extended configurations.
- (A3) In the retracted state, each shell is tightly contained in its parent (no space in-between).

Property (A1) is motivated by common manufacturing considerations, (A2) ensures deployability, and (A3) codifies the desire for compactness in the closed state. To develop a *geometric* characterization of shells, consider two closed connected sets $B \subseteq A \subseteq \mathbb{R}^3$, so that B and $\text{cl}(A \setminus B)$ represent the child and parent shell in a retracted

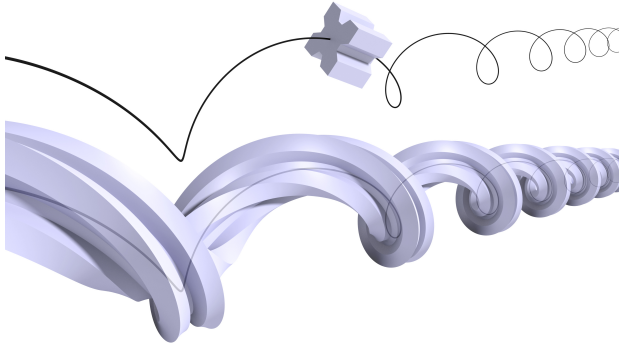


Fig. 2. The fundamental shape for a telescoping shell is a solid (top) swept along a helical path called the *medial curve*. Any such shape (bottom) can slide along itself via a continuous screw motion.

state (respectively); these sets satisfy (A3) by construction. Without loss of generality we can assume that A is fixed and B moves relative to A . We then seek geometries for B that allow (A1) and (A2) to hold, *i.e.*, that admit a feasible rigid trajectory between closed and open states. Any such motion must also satisfy these properties instantaneously: at each moment we must have an infinitesimal rigid motion that maps B to itself “almost everywhere,” *i.e.*, away from boundary points where the child makes contact with open space. Typically there is only one feasible infinitesimal motion, which we can express as a vector $\xi \in \mathfrak{se}(3)$, *i.e.*, a *screw motion* combining an infinitesimal translation and rotation. Until the child is separated entirely from its parent, constraints on its motion are identical; hence, the vector ξ remains fixed throughout some interval of time $0 \leq t \leq T$. Sweeping out any subset B along ξ then yields a helical solid $H := \bigcup_{t \in [0, T]} \exp(t\xi)B$, which is the generic shape for our telescoping shells (Fig. 2).

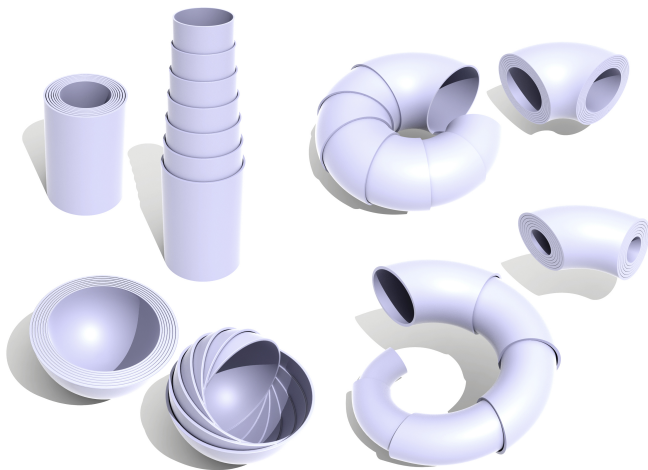
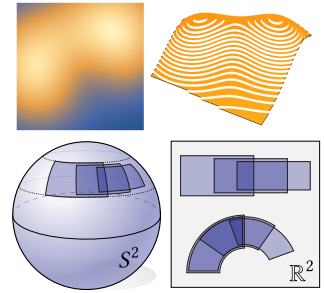


Fig. 3. A few basic axioms determine the possible shapes for a telescoping shell. Here we provide a classification for shells in \mathbb{R}^3 : linear, toroidal, helical (which subsumes the first two), and spherical (which we generally omit).

Classification of Telescoping Shells. In the case of pure translation, shells are linear; for pure rotation about an axis, they are toroidal (Fig. 3, top). For shapes with a high degree of symmetry there may also be more than one feasible screw motion, though these shells are not as useful for design—for instance, nested spherical shells admit any rotational trajectory (Fig. 3, bottom-left); stacks of planar slabs have two translational degrees of freedom. Noncompact shapes like the infinite helical tube in Fig. 2 also exhibit the desired motion, but do not have distinct “open” and “closed” states as required by (A2). Apart from these degenerate cases, the only shell geometries that satisfy our basic axioms are compact regions of \mathbb{R}^3 swept along helical trajectories (Fig. 3, bottom-right).

By no means do these axioms provide the only possible definition for a “telescope”—for instance, any foliation of a planar region can be used to generate a sort of telescoping structure (inset, top), albeit one that may not comprise a single sequence of shells; likewise, more designs are possible if we do not require that shells fit tightly inside each



other, at the cost of size and physical stability. One can consider telescopes that extend in alternating directions (Fig. 11b) or branch into multiple pieces (Fig. 11a), though the latter comes again at the cost of size. Finally, the axioms above can be used to define “telescopes” on other spaces: in the plane for instance, one obtains only circular or linear telescopes; on the sphere the only isometries are rotations, producing equatorial telescopes (inset, bottom).

Arbitrary Torsion. Though a helical shell with any cross section admits a telescoping motion (see inset), we will restrict our designs to *circular* cross sections. A crucial reason for this choice is that a circular shell in its fully-extended configuration can freely twist around its parent. Geometrically, this additional freedom provides impulses of *torsion* that greatly expand the space of possibilities beyond helices: in the limit of smaller and smaller shells, we can approximate curves with constant curvature and *arbitrary* torsion (Sec. 3.3).



For design and optimization we work primarily with the *medial curve* γ of the telescope (Fig. 2). The *fundamental theorem of space curves* states that any arc-length parameterized curve $\gamma(s) : \mathbb{R} \rightarrow \mathbb{R}^3$ is determined up to rigid motion by its curvature $\kappa(s) : \mathbb{R} \rightarrow \mathbb{R}$ and torsion $\tau(s) : \mathbb{R} \rightarrow \mathbb{R}$, *i.e.*, by integrating “bend” and “twist” along the curve, one can recover its position in space. (Concretely, one can recover γ by integrating the *Frenet-Serret formulas*.) Throughout, each shell will therefore be parameterized by a tuple of five scalars $S_i := (l_i, r_i, \kappa_i, \tau_i, \theta_i)$, where the first two determine the length and the radius (*i.e.*, “width”), respectively; the final parameter θ_i is a *torsional impulse* that will be discussed in Sec. 3.3.

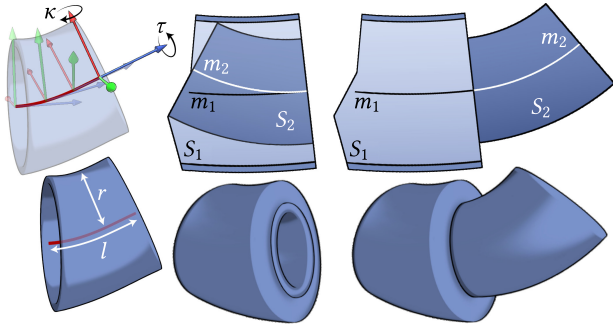


Fig. 4. Anatomy of a shell. *Left*: κ and τ determine the medial curve (in red); r and l determine the geometry of the shell surface. *Center, right*: retracted and open states of parent S_1 and child S_2 . In between, the motion slides the medial curve of S_2 along the endpoint of S_1 while keeping frames aligned.

3.2 Telescoping chains

We define a *telescoping chain* as a sequence of shells $\{S_0, \dots, S_n\}$. A chain satisfies the *nesting condition* if each child S_{i+1} is contained in its parent S_i when in the *retracted state*, *i.e.*, in the unique configuration where:

- The final points of the medial curves of parent and child are coincident, and
- The Frenet frames of the medial curves at their final points are also aligned.

(See Fig. 4, *center*.) A child S_i is *contained* in its parent S_{i-1} if every point of the parent is at least a distance $r_i + \epsilon$ from the medial axis of the child, where $\epsilon > 0$ is the wall thickness.

The extended configuration is given by aligning centerlines of consecutive shells such that the first point on the centerline m_{i+1} of S_{i+1} coincides with the last point on the centerline m_i of S_i , and the frames at these two points match. The telescoping motion is likewise generated by sliding S_{i+1} along the terminal point of S_i while keeping frames aligned.

For a given trajectory, we must also determine a suitable shell geometry. To determine radii r_i that satisfy the nesting condition we can iterate backward over the shells: for each shell S_i , compute the maximum distance d_{\max} between the medial curve of S_i and any point on the surface of the child shell S_{i+1} , then assign $r_i = d_{\max} + \epsilon$. Although we do not impose any hard restrictions on the values of l_i , κ_i , and τ_i , large variations in these values may have a detrimental effect on the compactness of the retracted structure. As seen in Fig. 4 and Fig. 7, a mismatch between consecutive shell profiles yields a large increase in the radius r_i , and regions of wasted space. Generally speaking, this behavior occurs whenever there is a large change in shell parameters. On the other hand, if l_i , κ_i , and τ_i are held constant, then the interior and exterior profiles are identical, and the radius need only increase by a minimal amount $\epsilon > 0$ corresponding to wall thickness.

Overall, the design of telescoping structures is therefore a balance between optimization of the telescope trajectory (to provide good geometric approximation), and uniformity of shell parameters (to encourage compactness).

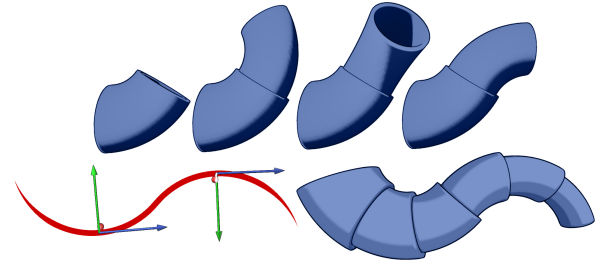


Fig. 5. *Top*: A unique feature of shells with *circular* cross sections is that in a fully-extended configuration they can be rotated relative to their parent. This additional degree of freedom greatly expands the space of possible shapes, such as the curve shown at *bottom*.

3.3 Torsional impulses

Shells with circular cross sections provide additional geometric flexibility by allowing each child S_i to rotate by an angle θ_i relative to its parent S_{i-1} around their shared tangent (Fig. 5, *top*). This additional rotation is applied only in the fully extended configuration (since otherwise there would be collisions; likewise, collisions preclude this kind of motion for non-circular cross sections). The curve in Fig. 5, *bottom* provides an illustrative example, requiring an instantaneous flip in the curvature normal at the inflection point. In this case we augment the telescoping motion by applying a “twist” at the end of every shell extension; as a result, only tangents (and not whole frames) will agree in the fully extended configuration.

Geometrically, this motion is encoded by impulses in the torsion τ of the medial curve. In particular, we get a torsion distribution $\tau(s) = \tau_0(s) + \sum_{i=1}^n \theta_i \delta_{s_i}(s)$ for some collection of impulse angles $\theta_1, \dots, \theta_n \in \mathbb{R}$ and parameters s_1, \dots, s_n along the curve (where δ denotes the Dirac delta distribution). Note that torsional impulses have no effect on the nesting condition—they simply add free parameters that facilitate better geometric approximation.

If κ_i and τ_i are identical for all shells then the extended path has continuous *scalar* curvature, but may not be G^2 since the curvature *normal* can jump. For general design tasks we sometimes relax this restriction (as in Fig. 4), in which case the curve is only G^1 .

Throughout the remainder we consider piecewise helical curves with circular cross sections and torsional impulses. Generalizations of the kind discussed in Sec. 3.1 provide interesting directions for future work.

4 PIECEWISE HELICAL APPROXIMATION

While piecewise helical curves are quite expressive, they can be difficult to manipulate directly. We therefore devise a method for approximating any given curve—while keeping in mind our secondary goal of compactness. A rough outline of our strategy is:

- (1) Approximate the given curve as a densely-sampled polyline.
- (2) Smooth its curvature via heat flow.
- (3) Partition it into segments and compute the best approximation of its torsion by a constant plus impulses.
- (4) Convert each segment into a telescoping shell.

Rather than work directly with vertex positions, optimization is framed directly in terms of (discrete) curvature and torsion. The

fundamental theorem of space curves ensures that a good approximation of these functions will yield a close approximation of the given curve geometry—while making it easier to satisfy the conditions needed for telescoping motion. Moreover, since there is no dependence of κ on τ (and vice versa), these functions can be optimized separately. We can avoid drift in the curve endpoints (which may need to connect to other curves) by finding a rotation and uniform scaling that aligns the endpoints of the helical approximation with the endpoints of the given curve.

4.1 Curve discretization

Given an arc-length parameterized input curve $\gamma(s)$, we first sample vertex positions $\gamma_0, \dots, \gamma_m$ at regular intervals of size $d > 0$. The associated (discrete) Frenet frame is then given at each vertex by

$$\mathbf{T}_i = \frac{e_{i+1}}{\|e_{i+1}\|} \quad \mathbf{B}_i = \frac{e_i \times e_{i+1}}{\|e_i \times e_{i+1}\|} \quad \mathbf{N}_i = \mathbf{B} \times \mathbf{T}$$

where $e_i := \gamma_{i+1} - \gamma_i$ is an edge vector. Curvature and torsion are then discretized as values κ_i, τ_i per vertex, equal to the rotation angles around the binormal and tangent (*resp.*) between previous and current frames. (If e_i and e_{i+1} are parallel we simply let $\mathbf{B}_i = \mathbf{B}_{i-1}$ so that $\tau_i = 0$.) When necessary, this data can be integrated to recover vertex positions γ_i , using the position and frame from the first point of the input curve as the initial data.

We next optimize the curvature and torsion functions on the finely-sampled curve; we then partition the curve into n helical segments which provide the centerlines for individual shells.

4.2 Curvature optimization

To obtain more uniform curvature, we apply a simple heat flow $\frac{d}{dt}\kappa = \frac{\partial^2}{\partial s^2}\kappa$ directly to the curvature function itself. The right-hand side is discretized via standard finite differences to obtain a matrix $L \in \mathbb{R}^{m \times m}$. We then use backward Euler to integrate the flow, via

$$(I - hL)\kappa^{k+1} = \kappa^k,$$

where I is the identity, h is the step size, and $\kappa^k \in \mathbb{R}^m$ is the vector of curvatures at step k . Terminating this flow before we arrive at a constant κ provides a tradeoff between uniformity and fidelity. To obtain the final curvature value for each helical segment, we simply take the average over all fine vertices contained in that segment.

4.3 Torsion optimization

Optimizing torsion is not as straightforward, since we must simultaneously determine helical torsions τ_i and twist angles θ_i . Directly minimizing the difference of torsion functions is not meaningful due to impulses; we instead consider the *cumulative* torsion function

$$T(s) = \int_0^s \tau(x) dx. \quad (1)$$

For our piecewise helical curve “with impulses” this function has a very particular shape: it is a (typically discontinuous) series of affine segments, as pictured in Fig. 6, *top left*. The slope of each segment determines the torsion of the corresponding shell, and the (signed) height of each discontinuity determines the twist angle in the extended configuration. The problem is then to find the best piecewise approximation $T^*(s)$ of the cumulative torsion $T_0(s)$ of the original curve.

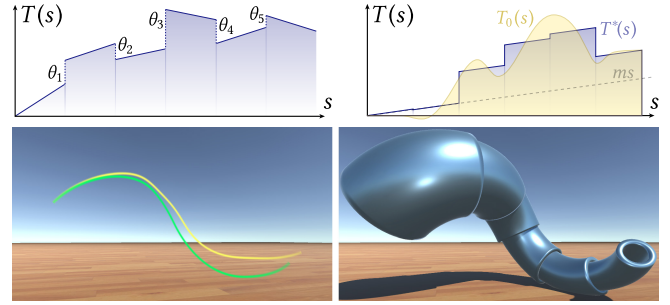


Fig. 6. *Top left*: The cumulative torsion function $T(s)$ for a helical curve with torsional impulses θ_i . *Top right*: Best approximation T^* of a given torsion function T_0 (using constant slope m), shown in yellow and green (*resp.*) on *bottom left*. *Bottom right*: resulting telescope.

Let p_0, \dots, p_n be the segment endpoints, with $p_0 = \gamma_0$ and $p_n = \gamma_m$; for $i = 1, \dots, n-1$ let m_i be the slope of the i th interval and let $\sigma_i = T^*(p_i)$ determine the height of the left endpoint. Any other point $s \in [p_i, p_{i+1}]$ can then be written as $T^*(s) = \sigma_i + m_i(s - p_i)$, giving us a per-segment torsion approximation error of

$$E_i = \int_{p_i}^{p_{i+1}} (T_0(x) - T^*(x))^2 dx,$$

whose sum defines an energy $E_{\text{fidelity}} = \sum_{i=0}^{n-1} E_i$. From a minimizer of this energy one can recover the impulses $\theta_1, \dots, \theta_{n-1}$ via

$$\theta_i = (\sigma_{i-1} + l m_{i-1}) - \sigma_i,$$

where l is the segment length and $\sigma_0 = 0$.

4.3.1 Complexity and Compactness. Minimizers of E_{fidelity} are typically unsatisfactory for two reasons. First, all impulse values θ_i are typically nonzero. In physical fabrication, torsional impulses typically demand additional complexity (*e.g.*, additional mechanical actuators or more elaborate shell geometry). We therefore add an L^1 term to encourage sparsity:

$$E_{\text{sparsity}} = \sum_{i=1}^{n-1} |\theta_i| \quad (2)$$

Second, there may be large variations in the segment torsions (determined by the slopes m_i), which reduces the compactness of the retracted configuration (Fig. 7). Here we either add a penalty

$$E_{\text{regularity}} = \sum_{i=1}^{n-1} (m_i - m_{i-1}), \quad (3)$$

or simply use a single degree of freedom m to control the slope of all segments (as in Fig. 6, *top right*). While the latter choice may seem restrictive, the torsional impulses alone can still do a surprisingly good job of approximating the given torsion function (as seen in most examples throughout). Our overall energy is then

$$E = E_{\text{fidelity}} + \alpha_i E_{\text{sparsity}} + \alpha_u E_{\text{regularity}} \quad (4)$$

where α_s and α_r influence the number of impulses and the compactness (*resp.*), and $E_{\text{regularity}}$ is often omitted in lieu of a single slope degree of freedom (m).

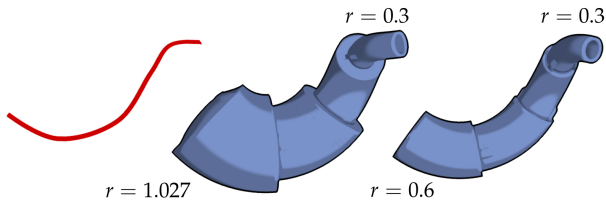


Fig. 7. Both telescopes approximate the constant-curvature curve on the left. *Middle*: Varying torsion on the shells leads to large shell sizes. *Right*: Using only impulses improves compactness while remaining nearly as accurate.

4.4 Telescope Generation

The only undetermined parameters remaining are the shell radii r_i , which determine the overall “thickness” of the telescope. Since the radius of each parent is uniquely determined by the radius of its child (Sec. 3.2), the only degree of freedom is the radius of the final, innermost shell. We allow this value to be specified by the user; if no radius is specified, a default value of twice the wall thickness ϵ is used. These radii will satisfy the nesting condition by construction. All that remains now is to generate the explicit shell geometry.

4.4.1 Curvature/torsion parameterization of helix. To construct centerlines, we need an expression for a helix with given curvature and torsion. Let $a = \kappa/(\kappa^2 + \tau^2)$, $b = \tau/(\kappa^2 + \tau^2)$, and $c = \sqrt{a^2 + b^2}$. One can then easily show that

$$h(t) = \left(a \cos \frac{t}{c}, a \sin \frac{t}{c}, \frac{bt}{c} \right)$$

is an arc-length parameterized helix with curvature κ and torsion τ .

4.4.2 Shell geometry generation. The exterior profile of shell S_i can now be expressed as

$$f(s, \varphi) = h(t) + r_i \cos(\varphi)N_h(t) + r_i \sin(\varphi)B_h(t),$$

for $s \in [0, l_i]$, where N_h and B_h denote the unit normal and binormal (*resp.*) of a helix h with torsion τ_i and curvature κ_i (in practice we sample this function at regular intervals to construct a mesh). For the inner profile we substitute the radius $r_i - \epsilon$ for r_i . Annular caps at either end yield a closed surface delineating the boundary of the solid shell.

3D printing considerations. Mathematically, child and parent shells make only tangential contact in the extended state. In practice, we use a linear tapering of shell radii (along their length) to keep shells connected; we also slightly extend the length of each shell. As a result, each shell gets “stuck” inside its parent, preventing the telescope from disconnecting. To realize given torsional impulses we carve a channel into the interior profile of each parent, and add small protrusions at the base of each child (Fig. 8, *left*). Together, this geometry guides the extension motion of the child, and allows the child to twist as far as necessary but no further. Finally, we add a small gap between consecutive shells, both to give room for extended shells to rotate, and to accommodate tolerances of 3D printers.

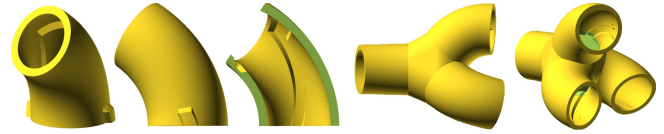


Fig. 8. *Left*: Shell geometry with protrusions and interior channels. The circular arc at the rim of the shell controls the rotation from the torsional impulse. *Right*: An example of a fused juncture piece.

5 NETWORKS AND JUNCTURES

To fabricate more interesting structures, we connect a network of telescoping chains via *junctions*. These junctions are simply fixed objects to which multiple curves are rigidly attached, as shown in Fig. 9; each junction has one “parent” curve, and any number of “child” curves. In our system, junctions are created in the initial spline drawing phase, and are preserved through all subsequent phases. Curve networks with tree topology ensure that our telescoping structures can always extend without locking, though in principle loops are possible (Fig. 11). Globally, all telescoping motion is expressed relative to some fixed root object.

5.1 Juncture geometry

We give junctures simple geometry; either (i) the convex hull of the bases of all incident chains, or (ii) a sphere with radius sufficient to contain all bases of incident chains (see inset), where the *base* of a chain is the circular cross section nearest to the juncture. One can of course substitute a custom mesh for any juncture, providing additional functionality or better geometric approximation. To ensure feasible motion, each retracted chain is subtracted from its incident junctures. In practice, juncture geometry unioned with the outermost parent of each incident chain (Fig. 8, *right*); to determine globally feasible attachment points for each chain we iteratively check and resolve collisions until there are none (see accompanying video).

5.2 Radius interpolation

Apart from the nesting condition, shell radius has a significant effect on the bulk geometry of the extended structure. Judicious choice of radii helps to further improve geometric approximation. In particular, for a chain attached to two junctions, we have target radii $r < R$ at the two endpoints (determined, for instance, by the input geometry). If L is the arc length of the centerline and m_t is the tapering slope, then $R - r - Lm_t$ is the amount by which the radius must decrease along the chain. Since shell radius decreases by ϵ per shell, we set the number of shells to

$$\max(\lceil (R - r - Lm_t)/\epsilon \rceil, 2)$$

which approximates the desired decrease to within ϵ . Smaller wall thickness yields more shells per chain, resulting in greater compactness; it also provides more torsional impulses, and hence better approximation of the given path (Fig. 10). Real fabrication processes of course dictate a minimum possible wall thickness (which can be mitigated by increasing the size of the structure itself).

6 MESH SKELETONIZATION

For complex geometry, it may be difficult or time consuming to design a curve network by hand. We partially automate this process via *mesh skeletonization*; any such method can be used as long as the final skeleton consists of 1D curves rather than 2D regions (as with the medial axis). We use a variation of the method of Tagliasacchi *et al.* (2012), where vertices are pushed along the normal direction until the mesh degenerates into such a skeleton. We then perform edge collapse until no triangles remain (only edges), using the *quadric error metric* to guide simplification (Garland and Heckbert 1997). The resulting polygonal curves are then converted into Catmull-Rom splines, and junctures are placed wherever multiple curves meet. This network represents a starting point for optional user cleaning and tweaking, after which the network can be run through the rest of our optimization pipeline. An overview of this process is shown in Fig. 9.

7 RESULTS

We prototype several possible use cases for telescoping structures across a variety of domains, using both virtual and physical models. Here we stick to purely telescoping structures, though using telescopes in conjunction with other deployable mechanisms (*e.g.*, folding or scissor joints) may also prove useful. Physical prototypes

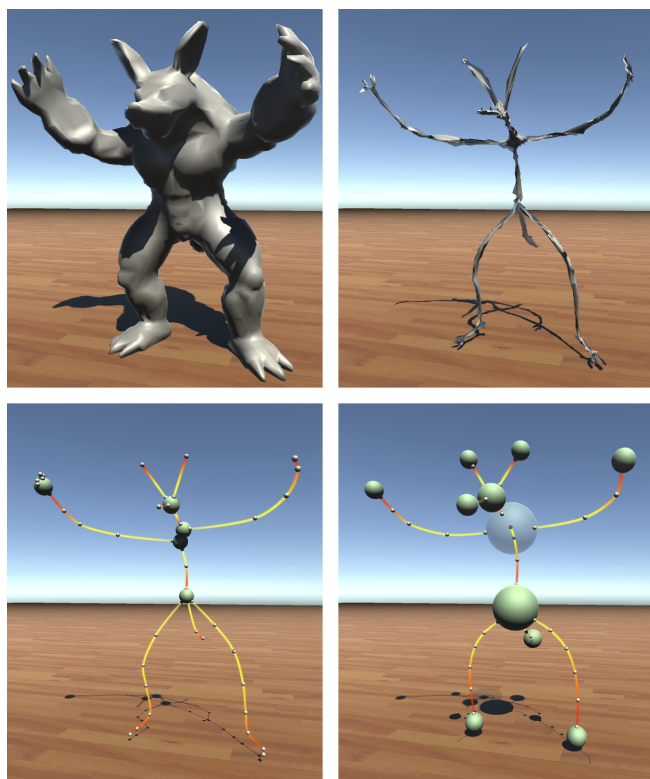


Fig. 9. Overview of the skeletonization process. The vertices of the original model are collapsed first to the skeleton and then to a spline network. After cleaning, the spline network can be discretized and ultimately converted to a telescoping structure.

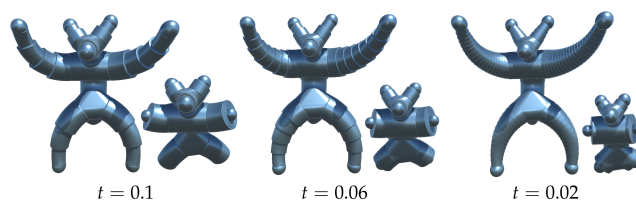


Fig. 10. Telescoping armadillos created from the same curve network, but with varying wall thicknesses t . Shell count increases as thickness decreases.

are shown in Fig. 12. We use basic consumer-level FDM 3D printing, which already allows for some fairly sophisticated designs (in spite of requiring a somewhat large wall thickness ϵ). Examples (a)–(c) show typical chains and junctures; Fig. 12c partially embeds the middle chain in the hull of the juncture. Fig. 12d shows the printed armadillo model from Fig. 9. Here one might modify the geometry of the junctures and outermost shells to better mimic the original surface appearance (*e.g.*, by adapting *bas relief* methods (Schüller *et al.* 2014)). Additional *simulated* examples are shown in Fig. 13a–n; the smaller relative wall thickness in these models likely demands a more sophisticated fabrication technique than FDM (or larger overall scale).

Fig. 13o–q shows the frame for a deployable shelter, attached to a flexible tarp; a larger, thinner version of such a shelter might be carried on one’s back and rapidly deployed at a camp site. In robotics, telescoping joints can be used to design vehicles that must pass through difficult terrain—as depicted in Fig. 13i–j where pieces of a vehicle retract in order to pass through a narrow passageway. Likewise, actuated torsional impulses would provide a robotic joint that is highly controllable, allowing (for instance) an end effector to be flexibly manipulated. To explore this idea, we implemented basic inverse kinematics (IK) where the difference between the endpoint position x and a goal position x^* is differentiated with respect to the impulse angles θ_i . This IK scheme is used in Fig. 13k–l to fetch objects with a telescoping claw, and in Fig. 13g–h to build a “hexapus” robot with six retractable arms which exhibit highly organic motion.

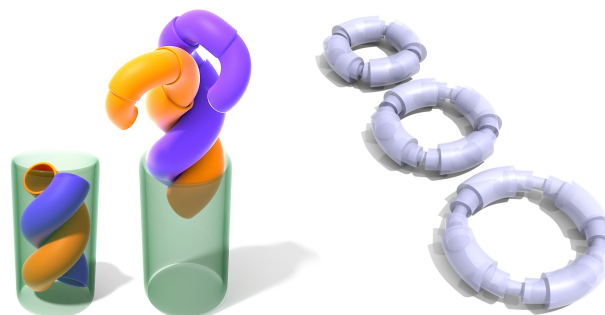


Fig. 11. Many possible generalizations remain to be explored. *Left*: a hypothetical “helical splitter” for connecting two telescoping chains. *Right*: a telescope with cyclical rather than tree-like structure.



Fig. 12. *Left:* Telescopes designed using our pipeline. *Right:* 3D printed physical prototypes.

8 FUTURE WORK

Our basic model for telescoping structures provides a jumping-off point for a variety of generalizations, some of which are discussed in Sec. 3.1. One is to improve the quality of geometric approximation, either by augmenting the appearance of the exterior surface (as discussed in the previous section), or by incorporating non-circular profiles in parts of the structure where torsion impulses are not needed. Likewise, we currently optimize only the extended shape—simultaneously optimizing the retracted geometry might provide more meaningful aesthetics or additional functionality in the retracted state (e.g., improved packability). Replacing rigid junctures with telescoping “splitters” (à la Fig. 11, left) would help to improve the compactness of the retracted form (at the cost of more rapid shrinking of shell size). It is also interesting to consider the conditions under which a cyclical telescope (Fig. 11, right) or network of such cycles admits feasible extension/contraction. Finally, mechanical actuation of extension and torsional impulses would yield automatic deployability and controllability, facilitating the aforementioned applications in engineering and robotics.

REFERENCES

- Byoungkwon An, S. Miyashita, M.T. Tolley, D.M. Aukes, L. Meeker, E.D. Demaine, M.L. Demaine, R.J. Wood, and D. Rus. 2014. An end-to-end approach to making self-folded 3D surface shapes by uniform heating. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. 1466–1473.
- Oscar Kin-Chung Au, Chiew-Lan Tai, Hung-Kuo Chu, Daniel Cohen-Or, and Tong-Yee Lee. 2008. Skeleton Extraction by Mesh Contraction. *ACM Trans. Graph.* 27, 3, Article 44 (Aug. 2008), 10 pages.
- Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete Elastic Rods. *ACM Trans. Graph.* 27, 3, Article 63 (Aug. 2008), 12 pages.
- Florence Bertails, Basile Audoly, Marie-Paule Cani, Bernard Querleux, Frédéric Leroy, and Jean-Luc Lévêque. 2006. Super-helices for Predicting the Dynamics of Natural Hair. In *ACM SIGGRAPH 2006 Papers (SIGGRAPH '06)*. ACM, New York, NY, USA, 1180–1187.
- Florence Bertails-Descoubes. 2012. Super-Clothoids. *Comput. Graph. Forum* 31, 2pt2 (May 2012), 509–518.
- Romain Casati and Florence Bertails-Descoubes. 2013. Super Space Clothoids. *ACM Trans. Graph.* 32, 4, Article 48 (July 2013), 12 pages.
- Kenneth Cheung, Tomohiro Tachi, Sam Calisch, and Koryo Miura. 2014. Origami interleaved tube cellular materials. *Smart Materials and Structures* 23, 9 (2014).
- Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, and Joëlle Thollot. 2013. Floating Tangents for Approximating Spatial Curves with G1 Piecewise Helices. *Comput. Aided Geom. Des.* 30, 5 (June 2013), 490–520.
- Tamal K. Dey and Jian Sun. 2006. Defining and Computing Curve-skeletons with Medial Geodesic Function. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing (SGP '06)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 143–152.
- Akash Garg, Alec Jacobson, and Eitan Grinspun. 2016. Computational Design of Reconfigurables. *ACM Trans. Graph.* 35, 4, Article 90 (July 2016), 14 pages.

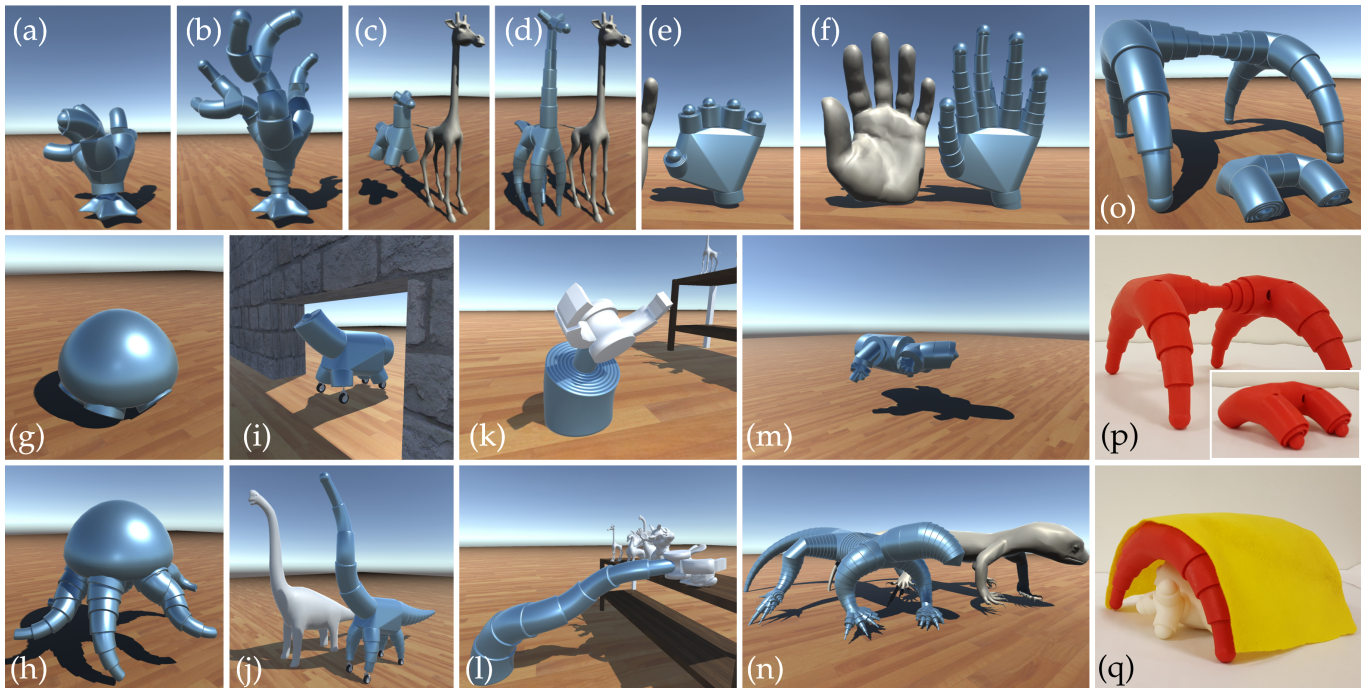


Fig. 13. Gallery of telescoping structures and applications. (a–b) Tree. (c–d) Giraffe. (e–f) Hand. (g–h) “Hexapus” with IK-driven tentacles. (i–j) Telescoping vehicular brachiosaurus. (k–l) Telescoping robotic crane driven by IK. (m–n) Lizard. (o–q) Deployable shelter housing the armadillo from Fig. 12.

Michael Garland and Paul S. Heckbert. 1997. Surface Simplification Using Quadric Error Metrics. In *Proc. SIGGRAPH*. 209–216. DOI : <https://doi.org/10.1145/258734.258849>

Charles H. Garrette and Harry M. Ryan. 1969. Telescoping Tube Assembly. (28 10 1969).

Mohammad Ghomi. 2007. h-Principles for curves and knots of constant curvature. *Geometriae Dedicata* 127, 1 (2007), 19–35.

Kaori Kuribayashi, Koichi Tsuchiya, Zhong You, Dacian Tomus, Minoru Umemoto, Takahiro Ito, and Masahiro Sasaki. 2006. Self-deployable origami stent grafts as a biomedical application of Ni-rich TiNi shape memory alloy foil. *Materials Science and Engineering: A* 419 (2006), 131–137. Issue 1–2.

Honghua Li, Ibraheem Alhashim, Hao Zhang, Ariel Shamir, and Daniel Cohen-Or. 2012. Stackabilization. *ACM Transactions on Graphics, (Proc. of SIGGRAPH Asia 2012)* 31, 6 (2012).

Honghua Li, Ruizhen Hu, Ibraheem Alhashim, and Hao Zhang. 2015. Foldabilizing Furniture. *ACM Transactions on Graphics, (Proc. of SIGGRAPH 2015)* 34, 4, Article 90 (2015).

Wilfred M. McCord and Walter J. Williford. 1966. Telescoping Pole. (08 11 1966).

J. Monterde. 2009. Salkowski Curves Revisited: A Family of Curves with Constant Curvature and Non-constant Torsion. *Comput. Aided Geom. Des.* 26, 3 (March 2009), 271–278.

S. Pellegrino. 2002. *Deployable Structures*. Springer Vienna.

L. Puig, A. Barton, and N. Rando. 2010. A review on large deployable structures for astrophysics missions. *Acta Astronautica* 67, 1&2 (2010), 12 – 26.

Roni Raab, Craig Gotsman, and Alla Sheffer. 2004. Virtual Woodwork: Making Toys from Geometric Models. *International Journal of Shape Modeling* 10, 01 (2004), 1–29.

L. Rhode-Barbarigos, N. Ali, R. Motro, and I. Smith. 2012. Design Aspects of a Deployable Tensegrity-Hollow-rope Footbridge. *International Journal of Space Structures* 27, 2-3 (2012), 81–96.

B. Salemi, M. Moll, and Wei-Min Shen. 2006. SUPERBOT: A Deployable, Multi-Functional, and Modular Self-Reconfigurable Robotic System. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. 3636–3641.

Greg Saul, Manfred Lau, Jun Mitani, and Takeo Igarashi. 2011. SketchChair: An All-in-one Chair Design System for End Users. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '11)*. ACM, New York, NY, USA, 73–80.

Christian Schüller, Daniele Panozzo, and Olga Sorkine-Hornung. 2014. Appearance-mimicking Surfaces. *ACM Trans. Graph.* 33, 6, Article 216 (Nov. 2014), 10 pages.

Liz Stinson. 2014. NASA Invents a Folding Solar Panel Inspired by Origami. (2014). <http://www.wired.com/2014/09/nasa-invents-folding-solar-panel-inspired-origami/> Online; posted September 22, 2014.

Tomohiro Tachi. 2010. Origamizing Polyhedral Surfaces. *IEEE Transactions on Visualization and Computer Graphics* 16, 2 (March 2010), 298–311.

Andrea Tagliasacchi, Ibraheem Alhashim, Matt Olson, and Hao Zhang. 2012. Mean Curvature Skeletons. *Comput. Graph. Forum* 31, 5 (Aug. 2012), 1735–1744.

Nobuyuki Umetani, Takeo Igarashi, and Niloy J. Mitra. 2012. Guided Exploration of Physically Valid Shapes for Furniture Design. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012)* 31, 4 (2012).

Kentaro Yasu and Masahiko Inami. 2012. POPAPY: Instant Paper Craft Made Up in a Microwave Oven. In *Proceedings of the 9th International Conference on Advances in Computer Entertainment (ACE'12)*. Springer-Verlag, Berlin, Heidelberg, 406–420.

Changxi Zheng, Timothy Sun, and Xiang Chen. 2016. Deployable 3D Linkages with Collision Avoidance. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '16)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 179–188.

Yang Zhou, Kangxue Yin, Hui Huang, Hao Zhang, Minglun Gong, and Daniel Cohen-Or. 2015. Generalized Cylinder Decomposition. *ACM Trans. Graph.* 34, 6, Article 171 (Oct. 2015), 14 pages.