Bend-It: Design and Fabrication of Kinetic Wire Characters

HONGYI XU*, Disney Research
ESPEN KNOOP*, Disney Research
STELIAN COROS, ETH Zurich
MORITZ BÄCHER, Disney Research

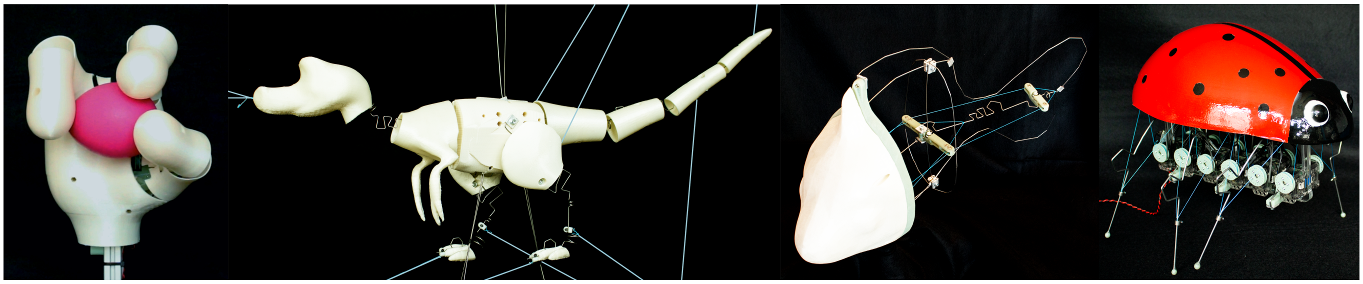


Fig. 1. We present a method for designing and fabricating kinetic wire characters, and showcase our technique on a set of examples including (from left to right) a soft robotic alien hand, an animated kinetic dinosaur, an animatronic fish with coupled wire contours, and a walking ladybug robot.

Elastically deforming wire structures are lightweight, durable, and can be bent within minutes using CNC bending machines. We present a computational technique for the design of kinetic wire characters, tailored for fabrication on consumer-grade hardware. Our technique takes as input a network of curves or a skeletal animation, then estimates a cable-driven, compliant wire structure which matches user-selected targets or keyframes as closely as possible. To enable large localized deformations, we shape wire into functional spring-like entities at a discrete set of locations. We first detect regions where changes to local stiffness properties are needed, then insert bendable entities of varying shape and size. To avoid a discrete optimization, we first optimize stiffness properties of generic, non-fabricable entities which capture well the behavior of our bendable designs. To co-optimize stiffness properties and cable forces, we formulate an equilibrium-constrained minimization problem, safeguarding against inelastic deformations. We demonstrate our method on six fabricated examples, showcasing rich behavior including large deformations and complex, spatial motion.

CCS Concepts: • **Computing methodologies** → *Physical simulation*;

Additional Key Words and Phrases: digital fabrication, computational design, wire bending, rods

ACM Reference format:
Hongyi Xu, Espen Knoop, Stelian Coros, and Moritz Bächer. 2018. Bend-It: Design and Fabrication of Kinetic Wire Characters. *ACM Trans. Graph.* 37, 6, Article 239 (November 2018), 15 pages.
https://doi.org/10.1145/3272127.3275089

1 INTRODUCTION

Computer-controlled bending machines transform straight metal rods into arbitrarily-shaped slender structures through mechanically-induced plastic deformations. This process of fashioning metal parts is fast, inexpensive, and with the recent introduction of desktop devices, it is now accessible to the general public. Wire bending therefore has the potential to become a viable alternative to 3D printing when it comes to creating physical representation of digital assets [Miguel et al. 2016]. As important added advantages, wire-bent structures exhibit strength and fatigue properties far exceeding those of 3D printed parts.

While bending and coiling operations reshape straight metal rods, they also govern the mechanical properties of the resulting designs, a principle that has been exploited for centuries. Steel coil springs, for example, are engineered to be much less stiff, and to have a much larger range of elastic deformations than the original rod they are composed of. The goal of our paper is to exploit this observation and enable the creation of a new class of bent wire structures designed to exhibit specific deformation behaviors.

Departing from the task of designing static wire structures, we present a computation-driven approach to creating *elastically deforming* kinetic characters from user-specified input such as an animation rig. Our cable-actuated or posable output models (see Fig. 1) well-approximate a set of user-specified target poses or keyframes under small actuation forces.

While a direct fabrication of an unoptimized wire network or rig is possible, local stresses would exceed the yield strength for all but trivial targets. While plastic deformation is essential during the fabrication process, inelastic deformation is intolerable after fabrication. Unique to kinetic wire characters is that stress can be relieved by locally bending the wire into spring-like entities with tunable homogenized stiffness. However, minimizing the maximal stress by naively extending the rest length of the wire leads to non-fabricable results with poor target matching performance due

*The first two authors contributed equally.

0730-0301/2018/11-ART239 $15.00
https://doi.org/10.1145/3272127.3275089

to local minima. Favoring the controlled placement of generalized springs, we face discrete questions of *where* to best introduce springs and *which* spring-like entity to use, with nonlinearities in both geometry and material behavior further complicating matters.

Overview & Contributions. To navigate this complex design space, we propose a novel, three-stage continuous optimization scheme, minimizing a target matching objective over three sets of design parameters:

(1) we first answer the binary question of *where* along the wire spring-like entities are needed by optimizing homogenized stiffness properties.
(2) To answer the discrete question of *which* spring-like entity to use, we replace detected regions with generic, non-fabricable entities which capture the behavior of our fabricable designs well, and optimize their parameters to calibrate their stiffness.
(3) To replace generic with fabricable entities, we compare their deformation behavior with a similarity metric, pre-building a database for fast retrieval of function-preserving, bendable types and corresponding parameters. To provide the user with fabricable options of varying volumetric footprint, we support behaviorally-redundant designs. While already matching targets closely, we refine retrieved parameters under strict fabricability constraints.

We co-optimize design parameters together with cable forces that act at a few, user-specified locations along the wire, constraining deformations to static equilibria and the elastic regime.

We have used our approach to design and fabricate a kinetic animatronic dinosaur, a stably walking robotic insect with 6 compliant wire legs, a soft robotic alien hand, a posable magnetic climber, compliant lamp designs and an animatronic fish with coupled wire contours. These demonstrations show the rich global behavior we can achieve by locally shaping the wire into spring-like entities of desired stiffness, size, and visual appeal, and furthermore validate our approach on input with large deformations and complex spatial motion.

2 RELATED WORK

Fabrication-Oriented Design. Triggered by advances in digital manufacturing technologies, the computer graphics community has contributed a significant body of work on fabrication-oriented design [Bermano et al. 2017]. A central focus in this body of work has been the development of computational tools that aid with design tasks which are challenging or tedious. Our work shares this overarching goal, and the computational methodology we present is tailored to fabrication via CNC bending machines. In contrast to previous work [Garg et al. 2014; Iarussi et al. 2015; Miguel et al. 2016], we target the design of *elastically deforming* kinetic wire structures. While not aimed at fabrication, Liu et al. [2017] propose an image-based reconstruction of wire structures whose output could serve as input to our technique. Metamaterial design [Panetta et al. 2017, 2015; Schumacher et al. 2015; Zhu et al. 2017] aims at achieving a desired macroscopic behavior by varying properties or structure at the microscopic level, relying on a single material for fabrication. We share this goal, however these works target solids where instead we target metal rod structures of constant radius.

Mechanism Design. The computer-assisted design of traditional mechanical assemblies has become a popular topic in graphics. Early work includes methods that aid with the understanding of complex assemblies [Mitra et al. 2010], and the design of mechanical toys and characters [Ceylan et al. 2013; Coros et al. 2013; Zhu et al. 2012]. Techniques for the computational design [Bächer et al. 2015; Megaro et al. 2014; Thomaszewski et al. 2014; Zheng et al. 2016] and editing [Bächer et al. 2015] of linkage-based assemblies followed. Recent approaches aid with the recovery [Lin et al. 2017] or retargeting [Song et al. 2017; Zhang et al. 2017] of existing mechanisms. Our work is most closely related to the computational design of compliant mechanisms [Megaro et al. 2017]. Like them, we safeguard against material failure and plastic deformation with a constraint on the von Mises stress. However, while they replace traditional mechanical joints with flexures and target 3D printing, we optimize stiffness properties of spring-like structures created by bending metal wire. This new problem setting demands a new approach for design optimization, as we will discuss shortly. Our work has applications in the design of lightweight robotic systems, complementing existing tools aimed at this exciting problem domain. Cellucci et al. [2017], for example, propose a system that enables bent wire robots to locomote. In our work, we leverage the compliance of the bent metal wires to design complex and expressive kinetic characters.

Compliant Rod and Cable-Actuated Structures. Pérez et al. [2015] describe a method to design flexible rod meshes, while Zehnder et al. [2016] propose a system for the design of structurally-sound ornamental curve networks. Like them, we rely on discrete elastic rods [Bergou et al. 2010, 2008] for accurate simulation. However, we target wire bending and not 3D printing, leading to a distinct and discrete design space which necessitates a different approach. For actuation, we optimize 3D forces generated by a sparse set of cables acting at user-specified locations along the wire structures. The actuation strategy we employ is inspired by prior work in graphics which has also used cables for actuation of deformable characters [Skouras et al. 2013], curved folded surfaces [Kilian et al. 2017], mechanical assemblies [Li et al. 2017; Megaro et al. 2017], and plush toys [Bern et al. 2017].

3 SIMULATING ELASTIC WIRE MODELS

To simulate elastically deforming wire, we use the discrete rod model introduced by Bergou et al. [2010; 2008]. Discretized into n nodes $\mathbf{x}_i$ and $n-1$ segments $\mathbf{e}^i = \mathbf{x}_{i+1} - \mathbf{x}_i$ (compare with Fig. 2 left), we use the standard stretch, bending, and twist potentials

$$E_s(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{n-1} k_s \varepsilon_i^2 \|\bar{\mathbf{e}}^i\|$$

$$E_b(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{n-2} \frac{1}{\bar{l}_i} (\boldsymbol{\kappa}_i - \bar{\boldsymbol{\kappa}}_i)^T \, \mathbf{B} \, (\boldsymbol{\kappa}_i - \bar{\boldsymbol{\kappa}}_i) \quad \text{with} \quad \mathbf{B} = \mathrm{diag}(k_b^{\mathbf{n}}, k_b^{\mathbf{b}})$$

$$E_t(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{n-2} k_t \frac{(\alpha_i - \bar{\alpha}_i)^2}{\bar{l}_i},$$

integrating the strain ε, curvature $\boldsymbol{\kappa} = [\kappa^{\mathbf{n}}, \kappa^{\mathbf{b}}]^T$, and twist α along the wire. Because the rest configuration of our bent wire is curved, we subtract the rest curvature $\bar{\boldsymbol{\kappa}}$ and twist $\bar{\alpha}$, dividing by the nodal integration domain $\bar{l}_i$.

Holding a sufficient set of degrees of freedom fixed, we simulate a piece of wire with unknown centerline nodes $\mathbf{x} \in \mathbb{R}^{3n}$ by minimizing the total energy

$$E(\mathbf{x}) = E_s(\mathbf{x}) + E_b(\mathbf{x}) + E_t(\mathbf{x}) - \mathbf{f}^T(\mathbf{x} - \bar{\mathbf{x}}),$$

subtracting the external work performed by forces $\mathbf{f} \in \mathbb{R}^{3n}$, to static equilibrium. Besides cable forces acting on a sparse set of nodes (in green in Fig. 2), external forces include the weight of rigid components attached to the wire and the wire's self-weight where needed. To avoid a dense Hessian, we minimize E with additional, per-edge rotation parameters, referring the interested reader to the original text [Bergou et al. 2010, 2008].

Our kinetic wire models consist of several pieces of wire, connected at their ends. To couple pairs of pieces, we introduce equality constraints $\mathbf{c}(\mathbf{x}) = \mathbf{0}$, asking the twist angles of the two end edges and their adjacent nodes to be equal. In contrast to a coupling energy [Pérez et al. 2015], our constraints ensure a complete transfer of forces and torques between neighboring pieces. We then minimize the Lagrangian

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = E(\mathbf{x}, \mathbf{f}) - \boldsymbol{\lambda}^T \mathbf{c}(\mathbf{x})$$

to first-order optimality.

We feed our bending machine with wire of constant radius r and cross-section $A = \pi r^2$, with stretch, bending, and twist stiffnesses

$$k_s = EA, \quad k_b^{\mathbf{n}} = k_b^{\mathbf{b}} = E\frac{Ar^2}{4}, \quad \text{and} \quad k_t = G\frac{Ar^2}{2}, \tag{1}$$

where G and E denote the shear and Young's moduli of the material. As we illustrate in Fig. 12, the standard elastic rod model predicts large, elastic deformations of kinetic wire models with high accuracy.

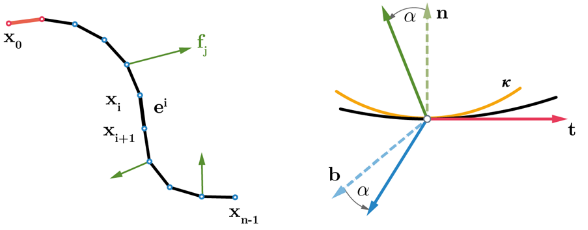


Fig. 2. We discretize our wire models with n nodes $\mathbf{x}_i$ and $n-1$ edges $\mathbf{e}^i$ (left), holding a sufficient set of degrees of freedom fixed (in red) when quasi-statically simulating its elastic behavior under a sparse set of cable forces $\mathbf{f}_j$ (in green). The angle α and vector $\boldsymbol{\kappa}$ measure the twist and curvature in the material frame $[\mathbf{t}, \mathbf{n}, \mathbf{b}]$ with tangent $\mathbf{t}$, normal $\mathbf{n}$, and binormal $\mathbf{b}$ (right).

4 DESIGNING KINETIC WIRE MODELS

Our method takes as input a user-specified initial rest configuration of a wire model. This wire model can either consist of a single or several connected wire pieces. These pieces in turn can either be curves drawn by the user with a spline tool, bones of a rigged character (see Fig. 3 Input), or contours extracted from a 3D model.

A user then selects a set of marker positions along the initial wire model and specifies one or several target positions for each of them (in blue in Fig. 3). These target positions $\hat{\mathbf{x}}^t$ can either be specified by selecting 3D positions directly, sampling points from a desired maker trajectory, or extracting them from a few keyframes for rigged input.

Our goal is then to change the rest configuration of the wire in such a way that the marker positions hit targets t as closely as possible under actuation forces $\mathbf{f}^t$ that vary per target (see Fig. 3 Output). Representing changes to the rest configuration with design parameters $\mathbf{p}$, we aim at minimizing the target matching objective

$$f_{\text{target}}(\mathbf{p}, \mathbf{f}) = \frac{1}{2}\sum_t \|\mathbf{S}\mathbf{x}^t(\mathbf{p}, \mathbf{f}^t) - \hat{\mathbf{x}}^t\|^2$$

where the selection matrix $\mathbf{S}$ extracts marker positions from the centerline nodes $\mathbf{x}^t \in \mathbb{R}^{3n}$, simulated to static equilibrium under forces $\mathbf{f}^t$.

For actuation, cables lend themselves. Note that we can match arbitrary targets under forces that act on a sufficiently dense set of locations along the wire. However, a large number of cables makes manual assembly an infeasible endeavor. While we could use a sparsity regularizer to select a small number of cables from a dense set [Skouras et al. 2013], we put our users in control and let them specify a set that acts at dedicated locations along the initial wire model. In doing so, we shift the design complexity to the controlled change of the model's rest configuration. The fewer cables we have at our disposal, the more challenging our design task. Driven by motors or pulled on by a user, the trade-off between force magnitude and the rest-length of the wire is an important design choice. With our cable force objective

$$f_{\text{cable}}(\mathbf{f}) = \frac{1}{2}\sum_t \|\mathbf{f}^t\|^2,$$

we provide the user with control over this trade-off.

While plasticity enables the fabrication of our wire models, it is important to safeguard against plastic deformations after fabrication, guaranteeing that our wire models return to their optimized rest configuration if cable forces are released. Following Megaro et al. [2017], we keep the maximum von Mises stress σ_{V} within cross-sections corresponding to centerline nodes $\mathbf{x}_i^t$, below the yield strength σ_{yield} of the material

$$\max_{t,i} \sigma_{\text{V}}(\mathbf{x}_i^t(\mathbf{p}, \mathbf{f}^t)) \leq \sigma_{\text{yield}}.$$

By varying design parameters $\mathbf{p}$, we aim at changing the wire model's rest configuration $\bar{\mathbf{x}}$. A naive minimization where we optimize $\bar{\mathbf{x}}$ directly, leads to poor target matching performance and non-fabricable output. We experimented with simple cycloidal and more general spline-based parameterizations, formulating regularizers using differential properties of the curve. However, for all but the simplest parameterizations, the target matching objective increases on the path from a straight to a wound up, spring-like shape. Hence, there is little hope to find a general parameterization that does not introduce local minima on the path to a desired rest configuration.

A key advantage of using wire is that we can locally increase the homogenized compliance by replacing wire segments with more wire, formed into a particular shape of small footprint such as, e.g.,

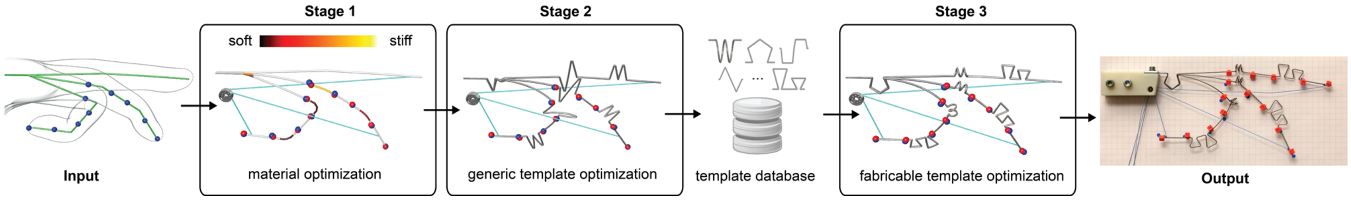


Fig. 3. **Designing a Kinetic Hand.** Our technique takes as input a rig or a user-specified initial rest configuration of a wire model, together with a set of target poses (Input). To match the input as closely as possible (Output), we then optimize for the positions and parameters of fabricable templates, along with cable actuation forces at a sparse set of locations. To avoid a prohibitively expensive, discrete optimization, we propose a three-stage design optimization: we first identify *where* to best place wire templates by optimizing local stiffness properties (Stage 1), restricting changes to a sparse set of regions. We then place generic templates in identified regions (Stage 2) and optimize their continuous parameters. Seeking best-matches in a database, we replace generic with fabricable templates and refine their parameters (Stage 3).

coils. To allow for varied and anisotropic deformation behavior (c.f. Fig. 7), we support several types of shapes, referring to them as *templates*. Each template is parameterized with a few continuous parameters (e.g., coil radius). If these parameters are within bounds that ensure their fabricability, we call them *fabricable templates*. Our aim is then to replace a sparse set of segments of our input with fabricable templates of varying size and shape.

However, while the parameters of a particular template are continuous, the question of *where* to place a template and *which* of our $|T|$ template types to use is discrete in nature. Representing the latter with discrete variables $p \in \{0, 1, \ldots, |T|\}$ where 0 denotes the no-template case, we face a design space of exponential complexity, prohibitively expensive to explore with a naive discrete optimization. The dependence of continuous template parameters on the values of variables p, further complicates matters.

To navigate this complex design task, we propose a three-stage design optimization where we minimize instances of the problem

$$\min_{\mathbf{p},\mathbf{f}} f(\mathbf{p},\mathbf{f}) + R(\mathbf{p}) \;\; \text{with} \;\; \mathbf{p}_{\text{lo}} \le \mathbf{p} \le \mathbf{p}_{\text{up}} \tag{2}$$

$$\text{subject to} \;\; \begin{bmatrix} \mathcal{L}_{\mathbf{x}}(\mathbf{x}^t, \boldsymbol{\lambda}^t) \\ \mathcal{L}_{\boldsymbol{\lambda}}(\mathbf{x}^t, \boldsymbol{\lambda}^t) \end{bmatrix} = \mathbf{0}, \;\; \forall t, \;\; \text{and} \;\; \max_i \sigma_v(\mathbf{x}^t_{t,i}) \le \sigma_{\text{yield}}$$

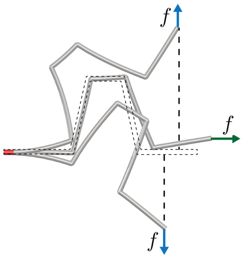
over three sets of design parameters $\mathbf{p}$, constraining simulated centerline nodes $\mathbf{x}^t$ to be static equilibria within the von Mises yield surface:

1. to identify places of *where* to best place templates (see Stage 1 in Fig. 3 and Sec. 4.1), we optimize the local stretch, bending, and twist stiffness along the initial rest configuration of the wire, adding a sparsity regularizer R to favor the stiffness properties to remain unchanged for most of the model. While this first stage does *not* change the rest configuration of the model, it answers the binary question of *where* to change $\bar{\mathbf{x}}$.
2. To answer the question of *which* template to use, we introduce a *generic template* (see Stage 2 in Fig. 3 and Sec. 4.2) which captures the behavior of our fabricable templates well, jointly optimizing the continuous parameters of several instances of this template, placed in regions identified with our first stage.
3. To replace generic with fabricable templates of matching behavior, we introduce a similarity metric to compare the deformation behavior of pairs of template instances, building a *template database* by sampling their parameters. In the third stage, we search this database for fabricable instances that closely approximate the behavior of our optimized generic templates (see Stage 3 in Fig. 3 and Sec. 4.3). Defaulting to the best match, we provide the user with the option to choose a template from a set of close matches. We then refine the template parameters further, leading to a kinetic wire model that is functional and ready for fabrication. With bounds $\mathbf{p}_{\text{lo}}$ and $\mathbf{p}_{\text{up}}$ on the values of template parameters, we ensure their fabricability.

Note that the discretization of the wire model changes at every stage of our design optimization. We update the selection matrix S and the attachment points of cables accordingly.

4.1 Identifying Sparse Template Regions

If we replace segments of the model's rest configuration $\bar{\mathbf{x}}$ with fabricable templates of varying shape and size, we essentially change the local stiffness properties in respective regions. Motivated by this observation, we first identify *where* along the wire changes to stiffness properties are necessary, separating them from regions where the wire can stay the same. In our second and third stage (Secs. 4.2 and 4.3), we will replace identified segments with fabricable templates, approximating the homogenized, non-physical behavior with changes to the rest configuration $\bar{\mathbf{x}}$ instead.



For wire with constant, circular cross-section, the stretch stiffness is high and the bending stiffness isotropic. However, when we replace a segment of a straight rod with an instance of a template, and apply forces to one end while holding the other one fixed (see inset), we observe a highly heterogeneous, anisotropic, and force-dependent behavior: if we apply an upward-pointing force (in blue), and apply the same force with opposite sign, we see an asymmetric deformation. If we apply a force along the direction of the straight rod (in green), we observe bending besides stretching.

Approximating the behavior of a kinetic wire model with an unknown set of templates, we optimize our generic problem (Eq. 2) with stiffness parameters $[k_s, k_b^{\mathbf{n}}, k_b^{\mathbf{b}}, k_t]$ which vary per edge *and* target $\mathbf{p}^{t,i}$, collected in the parameter vector $\mathbf{p}$. The bending stiffnesses in the normal $\mathbf{n}$ and binormal $\mathbf{b}$ directions are decoupled in order to support unrestricted anisotropy [Bergou et al. 2010, 2008].

Note that this optimization is bijective to locally estimating separate Young's moduli for stretch and bending, and a shear modulus for twist $[E_s, E_b^{\mathbf{n}}, E_b^{\mathbf{b}}, G_t]$. Replacing a wire segment with a template can only lead to an increase of compliance. To safeguard against local stiffening, we bound the stiffness parameters to be smaller or equal to the material properties $\hat{\mathbf{p}}$ of the wire with constant, circular cross-section (Eq. 1). To prevent them from becoming too soft, we bound them from below, thus keeping them within the physically-feasible range of fabricable templates

$$\beta\hat{\mathbf{p}} \leq \mathbf{p}^{t,i} \leq \hat{\mathbf{p}} \quad \text{with scale factor} \quad \beta < 1.$$

With our material optimization, we aim at answering the binary question of where stiffness changes are necessary. To restrict stiffness changes to a sparse set of regions, we penalize deviations from the physical properties $\hat{\mathbf{p}}$ across *all* targets with an approximate L_1-norm [Skouras et al. 2013]

$$R(\mathbf{p}) = w_{\text{sparse}} \sum_{i=1}^{n-1} \left(\sum_t \|\mathbf{p}^{t,i} - \hat{\mathbf{p}}\|^2 \right)^{\frac{1}{\alpha}} \quad \text{with} \quad \alpha > 2,$$

penalizing smaller values more strongly.

Note that the stretch stiffness is an edge-based quantity, while the bending and twist stiffnesses are node-based quantities [Bergou et al. 2010, 2008]. In simulations, we average stiffness parameters from neighboring edges to convert them to node-based quantities. If the estimated parameters $\mathbf{p}^{t,i}$ deviate from the physical properties $\hat{\mathbf{p}}$ by more than a factor of 0.2 for any of the targets t, we label the respective edge i for replacement. A key advantage of our material optimization is that we can detect these regions without having to change the rest configuration $\bar{\mathbf{x}}$ during minimization.

4.2 Placing and Sizing Generic Templates

Our end goal is to locally extend the rest-lengthof the wire in detected segments, shaping it into fabricable instances of $|T|$ templates in order to match our targets as closely as possible.

In experiments with our desktop bender, we found that limiting the templates' dimension along the wire to a constant length l is nonrestrictive, as multiple templates can be placed in series (see Fig. 4). Hence, we grow each local segment labeled for replacement, expanding it equally on either end until we can fit the smallest integral number of templates as we illustrate in Fig. 4 with a one- (left) and a two-template case (middle). If we — during an expansion — cross over to another segment, we merge the two, expanding them jointly (Fig. 4 right). Note that l is *not* measuring arc length but the distance between the two end points, connected with a straight line.

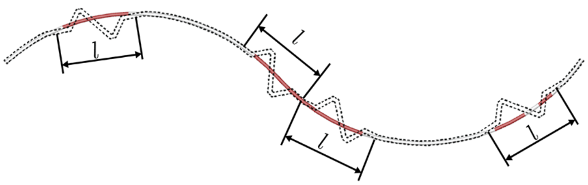


Fig. 4. We expand identified regions to fit the smallest number of discrete templates of length l.

With our material optimization, we significantly reduce the exponential design space by identifying the number m of locations where a template is needed. However, answering the question of *which* template to use at a particular location, is a formidable task, especially due to the coupling between template type and its parameters. Even if we had the computational capacity to minimize our design objective over the template parameters of all $|T|^m$ configurations, our design problem (Eq. 2) is highly non-linear and non-convex. As we will see in Sec. 4.3, parameter bounds that guarantee fabricability amplify the problem of getting stuck in local minima. Hence it is important to find a good initial guess before we enforce template-dependent fabricability constraints.

To avoid a discrete optimization, we seek a *generic template* whose parameters we can vary continuously, smoothly interpolating between desired stiffness properties, and representing our fabricable templates well. However, before we discuss our generic template, it is important to gain a high-level understanding of desktop CNC bending machines.

Desktop benders such as the DI Wire Plus or Pro have two degrees of freedom (DOFs) as we illustrate in Fig. 5: feed wheels control the extrusion of the wire (red) and the bend pin (green) is rotated to bend the wire. While we can pause bending and manually rotate the wire, spatial bends are restricted due to potential collisions between the wire and the machine, notably at the scale l of our templates.

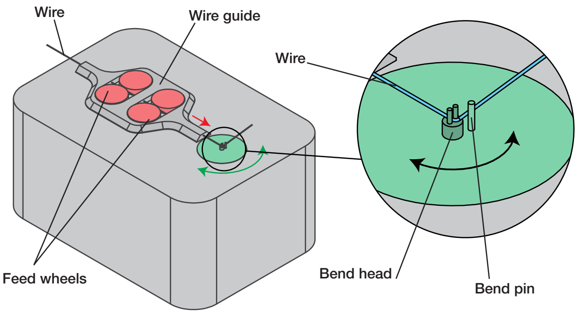
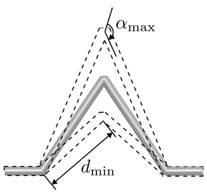


Fig. 5. Desktop benders have two degrees of freedom: feed wheels (red) control the extrusion of wire, while bends are made at the bend head by moving the bend pin (green). The bend pin can also be retracted, to enable clockwise and counterclockwise bends. While we can achieve spatial bends by pausing bending and manually rotating the wire, spatial bends are restricted due to collisions between the wire and the machine.

Low-end benders such as the DI Wire Plus cannot step the feed and bend wheels simultaneously, hence only support discrete bends. As will be discussed in Sec. 5, fabricable shapes are limited by a maximum bending angle $\alpha_{\max}$ and a minimum distance $d_{\min}$ between discrete bends, as shown in the inset on the left. Note that these fabrication constraints are universal. However, when switching devices or the wire, their values change.



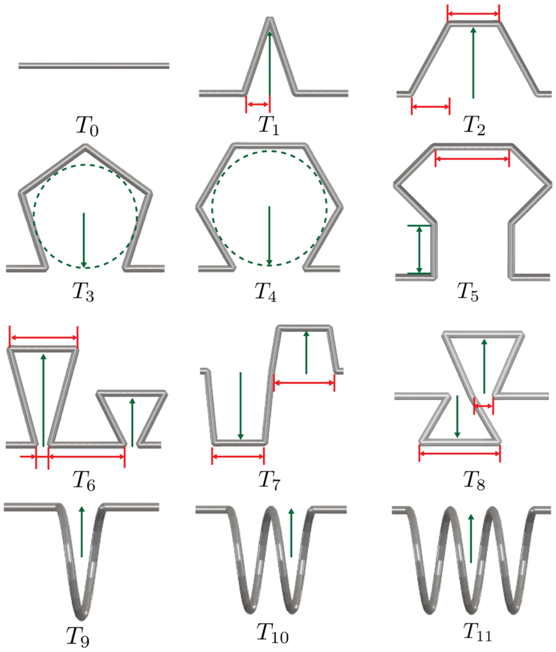


Fig. 6. **Fabricable Templates.** Besides the straight rod T_0, we support polygonal templates (T_1-T_5) which become more compliant with increasing number of sides. Our twin templates (T_6-T_8) consist of two isosceles trapezoids and further increase the stiffness gamut. Our helical springs (T_9-T_{10}) are not directly fabricable on the bending machine but are compact. Red parameters remain fixed when we fine-tune the green parameters.

Motivated by these fabrication considerations, we keep our fabricable templates, as well as our generic template, planar. To better approximate spatial behavior, we allow the local template planes to rotate about their wire axis, and optimize *three-dimensional* cable forces. To fabricate spatial kinetic wire models, we either pause and rotate the fed wire during, or assemble several planar pieces of wire after fabrication. As we demonstrate with our results and validations (see Sec. 6), locally planar templates are sufficient for creating a rich global deformation behavior.

A four-parameter template that fulfills our desiderata of genericity is shown in Fig. 8 left: partitioning the line between the two end points in 20 units, we define two isosceles triangles (in red) with constant base b and varying heights h_1 and h_2. With parameter w, we can calibrate the placement along, and with the angle θ the rotation about the template's axis. The genericity of our template becomes apparent when we vary the pairs of heights (h_1, h_2): if both of them are zero $(0, 0)$, we recover a straight line of length l. With a point-symmetric configuration $(h, -h)$, we control the magnitude of the stiffness along the wire. To fine-tune the anisotropy of the homogenized bending stiffness, we favor asymmetric configurations (compare middle, right in Fig. 8).

To find optimal parameters $[w, h_1, h_2, \theta]$ for our generic templates, we solve our design problem (Eq. 2) over a $4m$-vector $\mathbf{p}$, collecting all template parameters $\mathbf{p}_k$. We initialize both, the height parameters and rotation angles to zero, centering all templates by setting their offsets w to 3 units ($w = \frac{3}{20}l$).

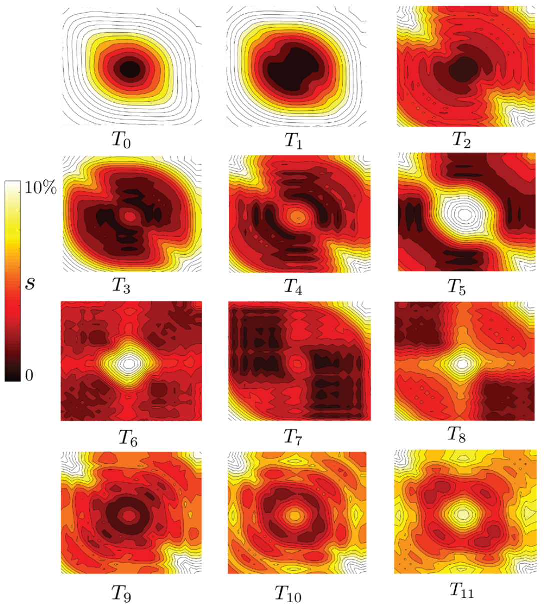


Fig. 7. **Similarity of Template Behavior.** Comparing the similarity s of generic to fabricable templates for varying height parameters, we observe polygonal templates (T_1-T_5) to well-approximate a concentrically increasing compliant behavior for small to medium height values. Twin templates well-capture highly compliant symmetric (T_6) and asymmetric (T_6-T_8) behavior, corresponding to large height values. While functionally overlapping with T_0-T_8 , the helical spring templates (T_9-T_{11}) have a small volumetric footprint for their stiffness range.

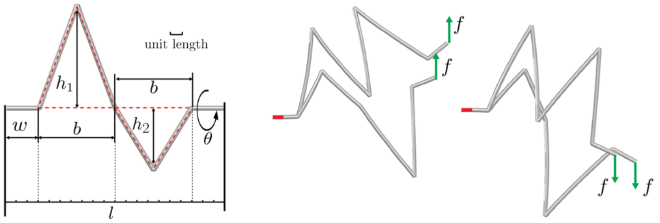


Fig. 8. **Generic Template.** Keeping the length l and triangles' base b constant, our four-parameter template (left) fulfills our desiderata of genericity. Asymmetric configurations allow us to control the anisotropy and magnitude of the homogenized bending stiffness (right).

With the two height parameters h_1 and h_2, we control the stiffness of the template. The larger their value, the more compliant the template. To prevent a template from becoming too soft, we bound the maximum allowable height from above and below

$$[0, -h_{\max}, -h_{\max}, -\pi] \leq \mathbf{p}_k \leq [l - 2b, h_{\max}, h_{\max}, \pi],$$

keeping the offsets and plane rotations within meaningful limits. For all our demonstrations, we set $h_{\max}$ to twice the length l, well-representing the stiffness gamut of our fabricable templates.

4.3 Replacing Generic with Fabricable Templates

While expressive and well-suited for iterative numerical optimization, our generic wire template is non-fabricable for a large and discontinuous subset of parameter values. In order to provide the user with fabricable alternatives of varying size and shape, we devise 12 templates, summarized in Fig. 6: templates T_0-T_8 are planar and can be fabricated on a low-cost bending machine. We also include helical spring templates (T_9-T_{11}) which are fabricated in a post-processing step using an easy-to-make, inexpensive hand tool (see Sec. 5). We describe our templates in more detail in the accompanying supplemental material.

As aforementioned, all our fabricable templates have constant overall length l. In addition to the parameters that allow adjustments to the lengths of a template's edges and the angles between them (in red and green in Fig. 6), we vary their spatial orientation with a rotation parameter θ, calibrating their placement within length l with an offset parameter w. Besides collisions, the two device- and wire-dependent parameters $\alpha_{\max}$ and $d_{\min}$ delimit the fabricability of our template designs. It is these fabricability constraints that motivate the explicit inclusion of template T_0 (not a subset of T_1) and templates with increasing number of sides: the maximum bending angle restricts the angle between neighboring sides, hence puts an upper bound on their lengths. The more sides we add, the more wire we can use for a template while keeping its footprint small. Intuitively, the more wire we can use, the softer we can make the homogenized stiffness of a template.

Comparing Templates. Stage 2 of our design optimization outputs parameter values $\mathbf{p}_G = [w, h_1, h_2, \theta]$ for m instances of our generic template T_G. For each instance, we seek to find a fabricable template T_I along with corresponding parameter values $\mathbf{p}_I$, such that the deformation behavior of the optimized generic instance is preserved.

To compare the behavior of a generic to a fabricable template, we initialize their rest configurations $\bar{\mathbf{x}}(\mathbf{p}_G)$ and $\bar{\mathbf{x}}(\mathbf{p}_I)$, and apply forces $\mathbf{f}_j$ to either end, keeping the other end fixed, resulting in deformed configurations $\mathbf{x}(\mathbf{p}_G, \mathbf{f}_j)$ and $\mathbf{x}(\mathbf{p}_I, \mathbf{f}_j)$. Selecting the deformed end nodes to which we apply the forces to with selection matrices S_G and S_I, we compute the root-mean-square (RMS) error of the differences between deformed end nodes, relative to the deformations of the ends of the generic instance

$$s(\mathbf{p}_G, \mathbf{p}_I) = \sqrt{\frac{1}{|F|} \sum_{j=1}^{|F|} \left(\frac{\|\mathrm{S}_{G,j}\mathbf{x}(\mathbf{p}_G, \mathbf{f}_j) - \mathrm{S}_{I,j}\mathbf{x}(\mathbf{p}_I, \mathbf{f}_j)\|}{\|\mathrm{S}_{G,j}\mathbf{x}(\mathbf{p}_G, \mathbf{f}_j) - \mathrm{S}_{G,j}\bar{\mathbf{x}}(\mathbf{p}_G)\|} \right)^2},$$

with small values of our *similarity metric* indicating a matching behavior. The RMS error means that larger deviations are penalized more. We apply the same $\frac{|F|}{2}$ in-plane and out-of-plane forces to either end, comparing the *spatial* behavior of our templates. Experimenting with various sizes of uniformly distributed forces, we found that a total of $|F| = 20$ forces are sufficient. Note that our similarity metric is asymmetric, reflecting our goal of an unidirectional replacement of a generic with a fabricable template.

Template Database. To efficiently replace a generic $\mathbf{p}_G$ with a fabricable instance $\mathbf{p}_I$, we pre-build a template database by uniformly sampling parameters and storing their pairwise similarity scores $s(\mathbf{p}_G, \mathbf{p}_I)$ for fast retrieval.

We sample template parameters within their feasible range, with the exception of the rotation parameter θ which is constrained to the two planar configurations $\theta = 0$ and $\theta = \pi$. For T_G and T_6, the two height parameters can be positive or negative, so we restrict sampling to $\theta = 0$ to avoid redundancy. For T_0, we omit parameter θ due to isotropy. After sampling, we simulate all instances of our generic (2, 187 samples) and fabricable (651 samples) templates, applying the same $|F|$ forces and computing their pairwise scores. Database construction only takes about 10 minutes and only needs to be constructed once for a particular wire radius.

Retrieving Templates. To retrieve function-preserving fabricable instances given an optimized generic template $\mathbf{p}_G$, we identify the K nearest generic samples $\mathbf{p}_G^k$ with distances d_k, setting the rotation θ to zero. For each nearest neighbor, we retrieve scores for our fabricable samples $\mathbf{p}_I$ and compute their weighted average

$$s(\mathbf{p}_G, \mathbf{p}_I) \approx \sum_{k=1}^{K} w_k s(\mathbf{p}_G^k, \mathbf{p}_I) \;\text{ with weights }\; w_k = \frac{\sum_j d_j - d_k}{(K-1)\sum_j d_j},$$

approximating the score for optimized values $\mathbf{p}_G$. Sorting them in ascending order, we present the L best matches to the user, letting them choose the aesthetically and functionally most pleasing fabricable instance $\mathbf{p}_I$. To account for spatial rotations $\theta \neq 0$ in the optimized parameters $\mathbf{p}_G$, we offset the retrieved rotation in $\mathbf{p}_I$ accordingly. For all our demonstrations, we use $K = 2$ and $L = 6$.

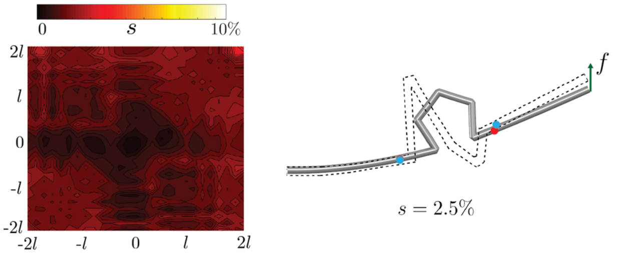


Fig. 9. **Similarity Scores.** Replacing optimized generic with fabricable templates, we can always find fabricable instances with scores of 2.5 % or less.

Varying the height parameters (h_1, h_2) of our generic template in their allowable range $[-h_{\max}, h_{\max}]$, we plot the contours of the minimum score across all types of templates T_I and corresponding parameter samples $\mathbf{p}_I$ in Fig. 9, restricting the minimum to samples drawn from a particular template type in Fig. 7. As we can see from Fig. 9, we can always find a fabricable instance with a similar *spatial* behavior for all combinations of heights, with approximation scores below 2.5 %. From Fig. 7, we learn that our polygonal templates with fewer sides (T_1,T_2) well-approximate the behavior for small height values, while templates with larger side counts (T_3-T_5) capture the behavior for medium values. Our twin templates (T_6-T_8) are well-suited to approximate symmetric (T_6) and asymmetric (T_6-T_8) configurations with large height values. While the functional behavior of our helical templates (T_9-T_{11}) is well-captured by planar templates (T_0-T_8), they have a significantly smaller volumetric footprint and are visually distinct. Overall, our database contains sufficient behavioral redundancy to provide the user with fabricable options of varying volume and appearance. However, a user could easily add custom template designs if desired.

As we illustrate in Fig. 10, our generic template approximates the spatial behavior of our fabricable templates well, with the approximation error for the majority of our fabricable template samples below 4 %.

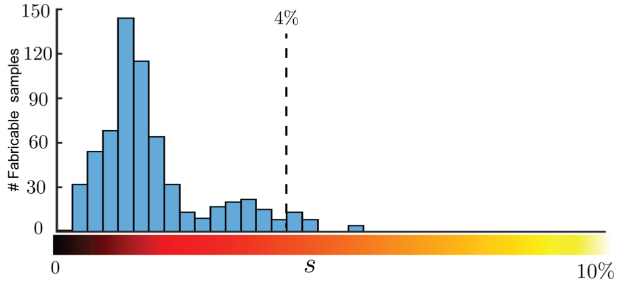


Fig. 10. **Approximation Error.** Our generic template captures the spatial behavior of our fabricable templates well, illustrated with a histogram of the scores for all fabricable samples. The approximation error is below 4 % for the majority of samples.

Refining Fabricable Templates. After our behavior-preserving replacement of the m generic templates, we have a fabrication-ready model that matches our targets well. To fine-tune the fabricable templates, we collect all parameters colored in green in Fig. 6 in a parameter vector $\mathbf{p}$ and solve a third instance of our design optimization (Eq. 2) under strict fabricability constraints. The red parameters (compare with Fig. 6) and offsets w remain fixed. Setting the initial guess to the parameter values retrieved from our database, we can further reduce the target matching objective. Note that a good initial guess is essential as fabrication constraints partition the design space into discontinuous regions.

4.4 Numerical Optimization

Besides our design parameters $\mathbf{p}$, we optimize cable forces $\mathbf{f}$. Note that cables can only be pulled on. Thus, we restrict their magnitude to non-negative values, optionally bounding them from above. While we keep the cables' attachment point on the wire, as well as the guide pulleys along the wire, fixed, we optimize the locations of the external pulleys, restricting them to user-defined regions in global coordinates.

We implicitly enforce static equilibria in Problem 2, relying on the implicit function theorem to compute the objective gradient

$$\frac{\partial f(\mathbf{y}^t,\mathbf{x}^t,\boldsymbol{\lambda}^t)}{\partial \mathbf{y}} + \frac{\partial f(\mathbf{y}^t,\mathbf{x}^t,\boldsymbol{\lambda}^t)}{\partial \mathbf{x}}\frac{\mathrm{d}\mathbf{x}^t(\mathbf{y}^t)}{\mathrm{d}\mathbf{y}} + \frac{\partial f(\mathbf{y}^t,\mathbf{x}^t,\boldsymbol{\lambda}^t)}{\partial \boldsymbol{\lambda}}\frac{\mathrm{d}\boldsymbol{\lambda}^t(\mathbf{y}^t)}{\mathrm{d}\mathbf{y}},$$

$$\begin{bmatrix} \mathcal{L}_{\mathbf{xx}} & \mathcal{L}_{\mathbf{x}\boldsymbol{\lambda}} \\ \mathcal{L}_{\boldsymbol{\lambda}\mathbf{x}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}\mathbf{x}^t(\mathbf{y}^t)}{\mathrm{d}\mathbf{y}} \\ \frac{\mathrm{d}\boldsymbol{\lambda}^t(\mathbf{y}^t)}{\mathrm{d}\mathbf{y}} \end{bmatrix} = - \begin{bmatrix} \frac{\partial \mathcal{L}_{\mathbf{x}}}{\partial \mathbf{y}} \\ \frac{\partial \mathcal{L}_{\boldsymbol{\lambda}}}{\partial \mathbf{y}} \end{bmatrix}$$

where we collect the design parameters $\mathbf{p}$ and cable forces $\mathbf{f}^t$ for target t in a vector $\mathbf{y}^t$ and omit the three arguments for all expressions involving the Lagrangian. Formulating an adjoint system, we compute the gradient by solving a *single* sparse system.

Note that the torsional strain $\boldsymbol{\varepsilon}_r$ remains zero for a circular wire. Stretch along the wire is insignificant and $\boldsymbol{\varepsilon}_s = 0$. The constraints on the yield strength [Megaro et al. 2017] thus simplify significantly, considering stresses caused by bending strain $\boldsymbol{\varepsilon}_b$ only.

For minimization, we use the standard interior point method and BFGS [Nocedal and Wright 2006] as implemented in the KNITRO package (residual tolerance: 10^{-6}, maximum number of iterations: 500). Turning a nonlinear, mixed discrete-continuous problem into three continuous problems, we avoid getting stuck in poor local minima and crucially reduce the time complexity of the overall problem. As we illustrate in Sec. 6 quantitatively and qualitatively, we observe good convergence and achieve low target matching error for all our demonstrations.

5 FABRICATION

For wire bending, we rely on a DI Wire Pro desktop CNC wire bender. To ensure consecutive bends are coplanar, we add additional plates around the bend head that constrain the wire from below and above. We also add a guide that prevents the wire from popping out of the bend head during complex close bends. The bending process along with the machine modifications can be seen in the accompanying video. After bending, individual wires are assembled using screws or crimped connectors.

Collisions & Calibration. For a wire shape to be fabricable, we need to avoid collisions during the fabrication. We observed that as the wire is compliant, non-local collisions and wire-wire collisions are not an issue: collisions result in small elastic deformations of the wire. We therefore only need to be concerned with local collisions, namely of the wire with the wire guide (see Fig. 5). Secondly, the distance between the bend head and bend pin restricts the feed length between consecutive bends. We enforce local collision avoidance by restricting bends to angles equal or less than $\alpha_{\max}$ and the minimum feed length with parameter $d_{\min}$. These constraints are taken into account in our database of templates, so all templates in the database can be fabricated. If the input rig has angles that are larger than $\alpha_{\max}$, we round off corners by introducing segments of length $d_{\min}$ in a pre-processing step.

We observed that during the first few deformations of the parts the bends would relax and bend angles would decrease by a few degrees. This is likely due to relaxation of the internal stresses in the wire. To account for this effect, we calibrate the machine so that the desired angle is reached after stresses have been relaxed. After fabrication, we manually agitate the bends to induce this relaxation. For improved accuracy, we compare the parts to reference drawings and make small manual corrections where required.

Although the DI Wire Pro supports the fabrication of continuous bends by simultaneously bending and extruding, we restrict ourselves to the least common denominator functionality of benders, enforcing universal fabrication constraints. Hence, our piecewise-linear shapes are fabricable with low-end (e.g., DI Wire Plus, $3,675) and industrial-grade CNC benders. Continuous bends would also greatly increase the complexity of calibrating the machine and thereby the fabrication error.

Planar vs. Spatial Structures. The DI Wire Pro is aimed at fabricating 2D planar wire shapes. We create 3D wire shapes by either: (1)

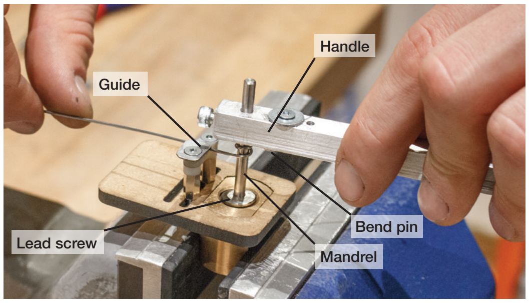


Fig. 11. We fabricate the coils for the spring-like templates using a simple and inexpensive hand tool. The mandrel size can be changed, allowing for coils with varying radii. The coil pitch is determined by the lead screw pitch.

Table 1. Template parameters and fabrication constraints for DI Wire Pro. All dimensions in mm. $\alpha_{\max} = 120°$ for all three radii.

r	l	$h_{\max}$	$d_{\min}$	Demonstrations
0.5	20	40	7	Dinosaur, Fish, Hand, Climber, Kinetic Finger
1	32	64	11	Ladybug, Lamp (small)
2	72	144	25	Lamp (large)

sequentially connecting multiple 2D wire shapes, controlling the rotation angle between wires; (2) pausing the machine and rotating the wire; or (3) manually adding out-of-plane bends as a post-process. In all cases, we use jigs to ensure the accuracy of the parts.

Coils. To increase the stiffness gamut, the database includes three coil springs (T_9, T_{10}, T_{11}). These are added in a manual post-process using a simple custom hand tool (cost < $20). The tool is shown in Fig. 11, and can also be seen in the accompanying video; creating a coil takes a few seconds. We add the length of wire required for the coil, so that after the coil is added the part has the correct dimensions. The tool works by wrapping the wire around a mandrel which is mounted on a lead screw, and the coil radius is varied by changing the radius of the mandrel.

6 RESULTS

We have used our technique to design and fabricate a total of eight kinetic wire structures (see Figs. 1, 12, 13, 14, 15, 16, 17, 18; accompanying video), illustrating applications in robotics, animatronics, and toy and furniture design, and kinetic art. To prevent the generic templates from becoming too soft, we set the scale factor β to 10^{-3} in Stage 1. For our sparsity regularization R, we use $\alpha = 4$. For minimization, we rely on KNITRO (interior-point method).

Fabrication. Depending on the targeted application or dimensions, we use spring steel wire (E = 209 GPa, G = 80 GPa) of radius 0.5, 1, or 2 mm and yield strength σ_{yield} = 1.6 GPa. The values for template length l, maximum height $h_{\max}$, and minimum side length $d_{\min}$ are radius-dependent and summarized in Tab. 1. The maximum bend angle $\alpha_{\max}$ is 120°, independent of the chosen wire radius.

Validation. To validate our three-stage design optimization, we compare simulations of a kinetic finger design to the fabricated result (see Fig. 12). To this end, we overlay the rest- and four simulated target poses with the fabricated finger, pulled on manually. As we can see in Fig. 12 (Poses 1–4), our actuated finger matches simulations well for target poses. For the rest shape of the finger, there are small deviations which are due to inaccuracies in the fabrication process (most likely due to the stress relaxation). However, unlike a finger design without templates (c.f. accompanying video), the fabricated finger only deforms *elastically* and goes back to its rest configuration when releasing the cable.

Kinetic Dinosaur. For our Kinetic Dinosaur, a rigged character with a full walking cycle served as input. We chose corresponding target points on 4 keyframes (see Fig. 13 left column) and estimated a kinetic character consisting of 4 pieces of wire with 3–7 templates each, actuated with a total of 10 cables (2 act on the neck, 2 on the tail, and 3 on each leg). During our three-stage optimization, we minimize our target matching objective by changing the rest shape of the wire, co-optimizing cable force magnitudes and pulley locations. We constrain the latter to a rectangular region and ask the cable forces to stay below 10N (maximum magnitude supported by motors). For the remaining frames, we only optimize force magnitudes, keeping pulley locations and the wire's rest configuration fixed. Actuating cables with position-controlled servo motors, the walking cycle of our physical character (wire model size: 60 cm ×20 cm ×7 cm) resembles the input animation well. We keep the body of the dinosaur fixed in global coordinates, optimizing the neck, tail, and legs that are rigidly connected to the body. At 1.7 mm (compare blue and red spheres in Fig. 13 bottom row), the mean target matching error is small. The maximum error is 4.8 mm. Over the complete walking cycle, the mean error is only 2.1 mm. From the input geometry, we generated disconnected shells that we rigidly attach to the wire.

Walking Ladybug. Starting from a curved, spline-specified leg design together with a desired end effector trajectory (see Fig. 14 Input, in blue), we optimize the rest configuration of a leg by inserting templates and optimizing forces of two attached cables. For target matching, we use 8 trajectory samples (see Fig. 14 Pose 1-3). While both planar, the desired trajectory and the leg at rest lie in *different* planes, providing an illustration of the rich *spatial* motion we can achieve with a planar piece of wire. Compensating for the weight of the shell, the frame, and the motors as also the contact forces acting on the end effector when a leg is in touch with the ground, we generate a stably walking, 6-legged robot (30 cm ×20 cm ×25 cm). At 2.9 mm and 4.1 mm, the mean and maximum trajectory errors are small and the optimized trajectory (in red) matches the desired trajectory well. Our Walking Ladybug is lightweight, inexpensive, and durable. We use thermoforming [Schüller et al. 2016] to create the shell.

Robotic Alien Hand. From an animated, rigged hand, we designed and fabricated a compliant robotic hand with three fingers (Fig. 15; thumb: 12 cm; other two fingers: 16.5 cm). Placing servos in the rigid palm of the hand, we control the motion of the fingers with cables that are routed along the fingers with three pulleys. Extracting targets from 5 keyframes from a grasping animation, our thumb's motion is spatial whereas the motion of the other two fingers is nearly planar. As for our Kinetic Dinosaur, we print and rigidly attach lightweight shells. Actuated with a single cable, our wire

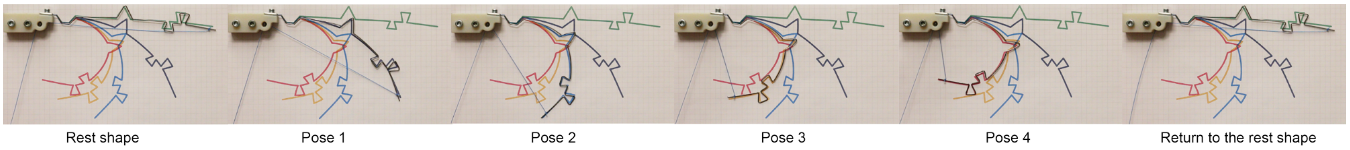


Fig. 12. **Validation.** A physical finger pulled on with optimized cable forces matches simulations closely (shown on the background paper with 5 mm grid spacing). Even after large deformations (target in red), our optimized kinetic finger does not plastically deform (Rest shape vs. Return to rest shape). Images are taken from a single continuous video sequence, which is included in the accompanying video.

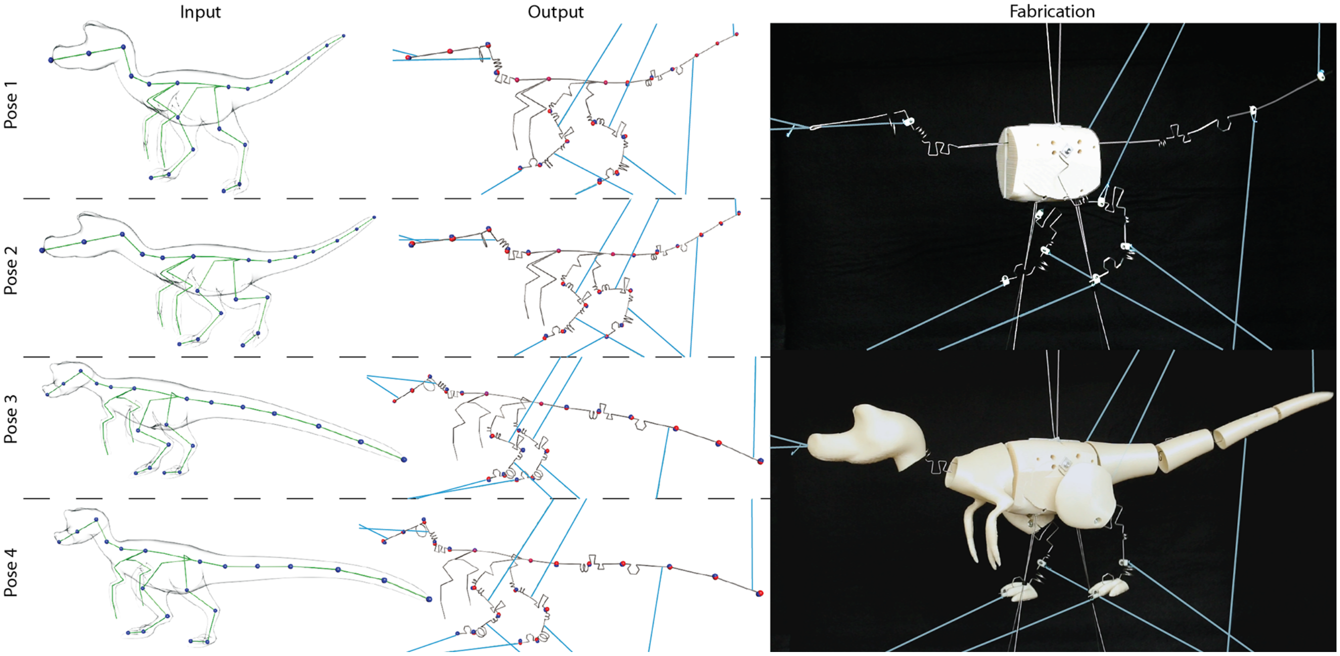


Fig. 13. **Kinetic Dinosaur.** We extract 4 keyframes from a walking cycle of a rigged dinosaur (Input, left column), matching corresponding target points (Output, middle column, in blue) on the animated rig by co-optimizing the wire's rest shape (red markers) and cable forces. The fabricated character with (bottom) and without (top) lightweight, 3D printed shells is shown in the right column.

fingers and cables remain in the enclosing shells during grasping. As we illustrate in Fig. 15 Fabrication with grasps of an arm and different sized balloons, interactions with our Robotic Alien Hand are safe. At 4.0 mm across all 5 targets, the maximum target matching error is small.

Magnetic Climber. From four poses of a character climbing over a wall, we generate a spatial wire structure (25 cm ×25 cm ×20 cm) consisting of 5 wire pieces (see Fig. 16). Our input poses are spatial and the rig is hierarchical. Enforcing coupling constraints between pairs of pieces as discussed, we add positional constraints at the hands and feet (2 magnets per limb), and 2 cables on the body . The spine and limb pieces undergo large, spatial deformations involving twisting. Besides the wire's rest shape, we optimize cable forces and attachment points, constraining the latter to lie in a plane. We constrain hands and feet to the same plane, adding box constraints to keep the locations of hands and feet within a small distance of their input locations. With a mean error of 3.0 mm and maximum error of 7.6 mm across all 4 target poses, the matching performance is high, given the complex, spatial behavior we aim to approximate. We add a wire head to the physical character (see Fig. 16 Fabrication), observing good agreement between simulated and physical poses.

Compliant Lamps. To illustrate applications in furniture design, we designed and fabricated two lamps (see Fig. 17, small: 30 cm; large: 75 cm). As input, the user specified a curved shape with a spline tool. Targeting a linear trajectory for the light bulb (in blue), we place and size templates and optimize the force magnitude of a single cable, routed along the wire with pulleys. We parameterize the last pulley position, constraining it to a plane parallel to the ground. In simulations and our optimization, we compensate for the self-weight of the rod and weight of the light bulb. After optimization, our small lamp can track the linear trajectory with a mean error of 4.4 mm. The large lamp tracks the targeted trajectory with mean error of 10.3 mm. At both scales, the mean error is below 1.5% of the size of the lamp. Using wire of different radii (see Tab. 1), our Compliant Lamps illustrate how we can use the same bending machine to create wire structures targeting a wide range of scales.

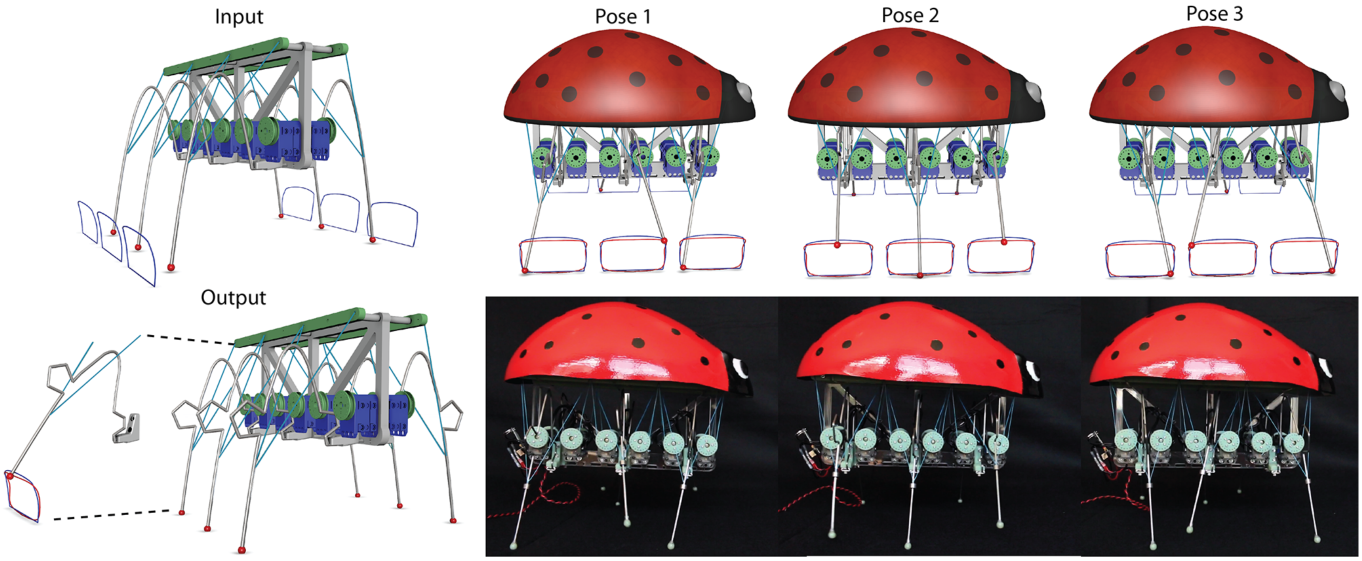


Fig. 14. **Walking Ladybug.** We optimized a spline-specified, curved leg (Input) to perform a desired end effector trajectory when actuated with two cables (Output, compare red and blue trajectories). Visualizing simulations for 3 out of 8 trajectory samples (Pose 1-3), we observe excellent target matching performance. We compensate for the self-weight of the robot and contact with the ground during our three-stage optimization, resulting in a stably walking robot.

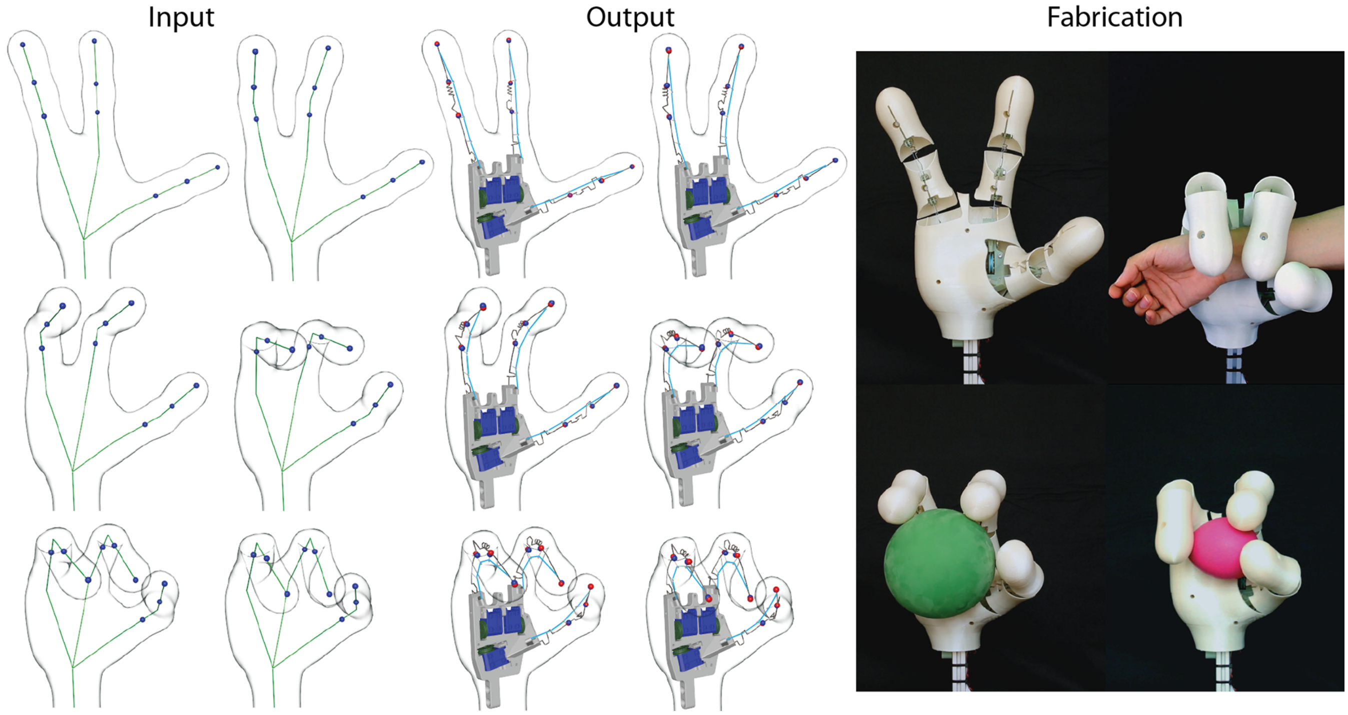


Fig. 15. **Robotic Alien Hand.** We turn a grasping animation (Input) of a hand with three fingers into a lightweight, compliant robotic hand (Output). We match corresponding targets (in blue) from 5 keyframes well (in red). Interactions with the hand are safe as we illustrate with grasping an arm and fragile objects like balloons of different sizes.

Animatronic Fish. For our last demonstration, we designed and fabricated an elastically deforming Animatronic Fish (see Fig. 18, 40 cm × 10 cm × 22 cm), given a swimming cycle of a rigged fish as

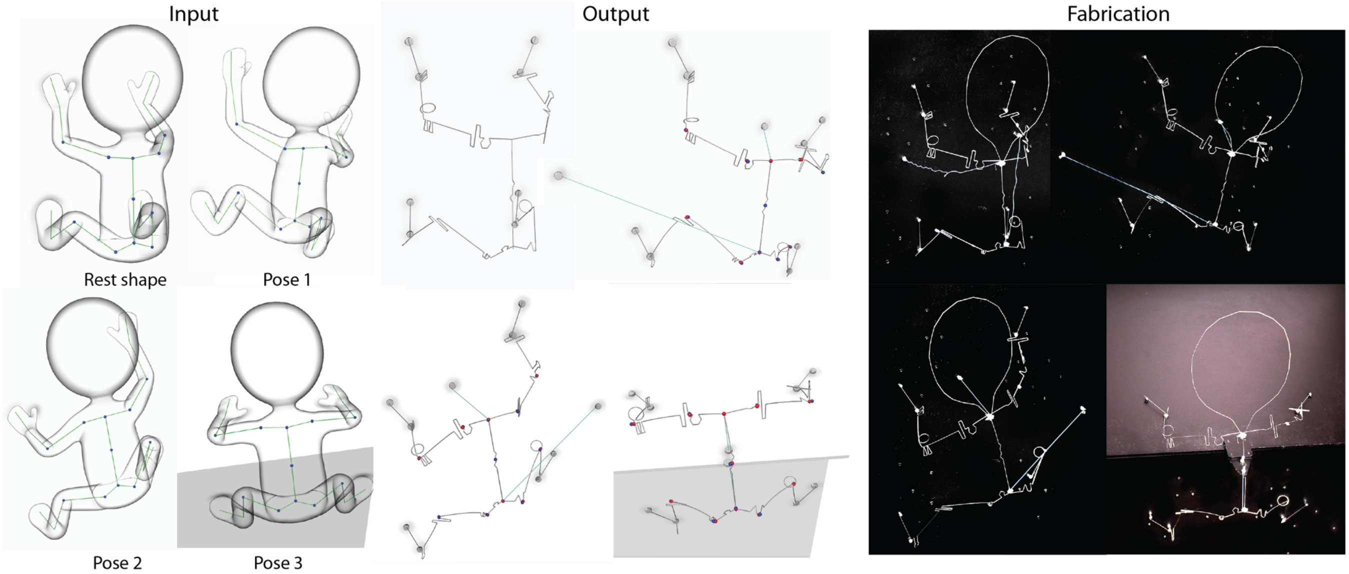


Fig. 16. **Magnetic Climber.** Our designed wire character matches a rest pose and 3 target poses, when attached to the wall using two magnets per limb along with 2 cables.

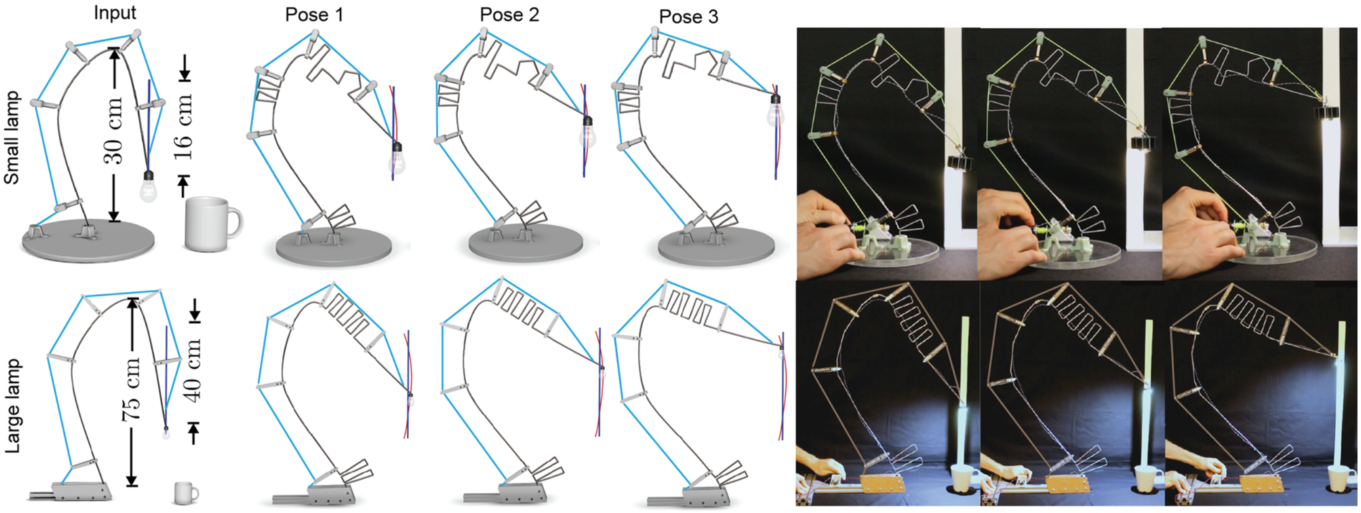


Fig. 17. **Compliant Lamps.** Using wire of 2 and 4 mm diameter respectively, we designed and fabricated a small and a large lamp. Closely matching the targeted linear trajectories (in blue), we compensate for the self-weight of the wire and the weight of the attached light bulb.

input. We extract the spine, body, fins, and traversal contours from the fish character in its bind pose, coupling 6 rods at a total of 9 locations. Driven by only four cables, we place and size fabricable templates along the spine of the fish. Keeping the head fixed in global coordinates, we constrain pulley locations to the interface between head and body. To avoid cables to collide with the transversal contours, we add additional pulleys on the spine. As we can see from Fig. 18 and the accompanying, our kinetic wire character approximates the graceful, continuous motion of the input animation very well. Our lightweight robot has applications in designing robotic fish [Katzschmann et al. 2018] besides animatronics. At 3.3 mm, the mean target matching error is low.

Comparison. Fig. 19 shows a set of comparisons, demonstrating the performance and necessity of our three-stage optimization. In (a), we demonstrate that local stiffness change is necessary for better tracking performance and stress reduction and avoidance of plastic deformation. In (b), we show that our optimized dinosaur leg (right) has significantly improved target matching performance, compared to a manually-designed leg (left) and a version optimized with Stages

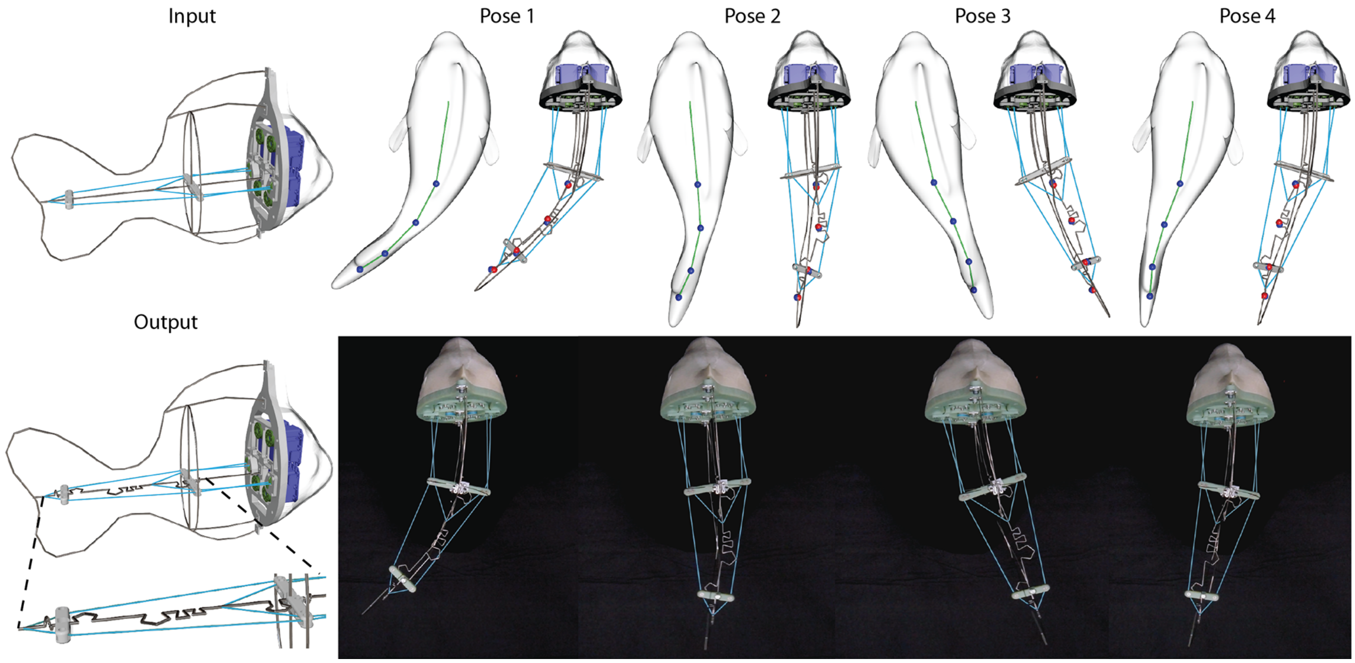


Fig. 18. **Animatronic Fish.** The input and optimized wire structure (on spine) are shown on the left from the side view. The first row also shows the 4 target poses (top view), from the input animation driven by the skeleton (in green) and output animation driven by the cables on the spine. The second row presents the frames for the swimming fabricated fish. The deformations are large and nonlinear.

1 and 3 (middle) where we replace detected regions with helical springs (T_9), skipping our generic template optimization (Stage 2).

One might argue that after Stage 1, we could run an exhaustive search, optimizing parameters for all combinations of fabricable templates. However, because an optimization of a single combination takes approx. 11 mins, an exhaustive search (for the dinosaur leg this would be 12^7 combinations as there are 12 fabricable templates and 7 template locations) is not tractable. Furthermore, our generic template optimization provides us with an initial guess for Stage 3 that is close to the solution, preventing us from getting stuck in a local minimum.

In (c) and (d), we compare unoptimized (left) to optimized trajectories (right), underlining the significance of the tracking performance we can achieve with our technique. Note that the input for both, our Ladybug leg and our lamp design, are smooth curves. Compared to rigged input, it is counter-intuitive to predict where along the wire fabricable templates are needed. Even for skeletal input, the material optimization does not exclusively place templates at or near rig joints (see, e.g., tail of our Dinosaur or spine of our Fish). In summary, our approach ensures fabricability, avoids plastic deformation, and significantly improve tracking performance.

Performance. Our simulation and three-stage optimization are performed on a machine with an Intel Core i7-7700 processor (4 cores, 4.2GHz) with 32 GB of RAM. We report the performance of our three-stage optimization for our kinetic finger design (1 cable, 3 templates) and a leg of our Kinetic Dinosaur (3 cables, 7 templates) in Tab. 2, with the mean errors $\bar{\epsilon}$ and maximum errors $\epsilon_{\max}$ computed across all target points and poses. Bending the finger takes 75 s on the DI Wire Pro, and the leg is bent in 87 s (excluding coil templates).

Table 2. Key statistics for the three optimization stages (material, generic template, and fabricable template optimization).

Example	opt. stage	#nodes	time (mins)	#iters	$\bar{\epsilon}$ (mm)	$\epsilon_{\max}$ (mm)
Dino leg	mat.	50	2.1	357	2.1	0.8
	gen. temp.	253	5.6	192	5.1	2.4
	fab. temp.	300	11.2	165	3.9	2.1
finger	mat.	46	1.3	148	1.9	0.7
	gen. temp.	130	3.1	106	2.8	1.6
	fab. temp.	127	1.4	51	2.6	1.5

7 CONCLUSION

We have presented a method for automatically generating kinetic wire characters, given an input animation rig or a network of curves along with target points. The deformation behavior of the wire is controlled by introducing spring-like templates at discrete locations along the wire, and we solve this design problem using a novel three-stage optimization approach. In the first step, we keep the wire geometry unchanged and determine sparse regions in which the wire stiffness should change. Next, we insert a generic template shape which closely captures the deformation behavior of fabricable templates, while providing a smooth optimization space. Finally,

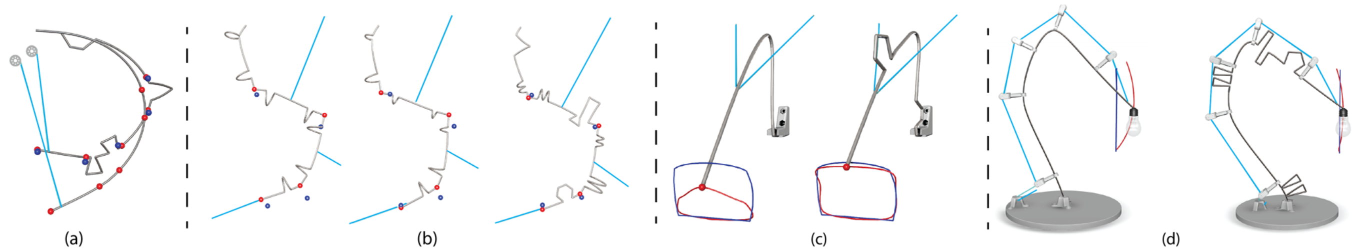


Fig. 19. **Comparisons**. (a) We compare our kinetic finger design to a piecewise linear rod (input). The piecewise linear rod has significantly worse tracking performance ($\epsilon_{\max} = 34.8$ mm), compared to our design ($\epsilon_{\max} = 2.6$ mm). This is largely due to the plastic deformation constraint. (b) Left: a manually-designed dinosaur leg where we place templates close to the rig joints (T_9, radius 5 mm); $\epsilon_{\max} = 11.7$ mm. Middle: we skip the generic template optimization and insert fabricable templates (T_9) with random initial parameters in regions detected with our material optimization; $\epsilon_{\max} = 10.3$ mm. Right: our Dinosaur leg design optimized with all three stages; $\epsilon_{\max} = 3.9$ mm. (c) Left: input ladybug leg which tracks the trajectory with $\epsilon_{\max} = 22.2$ mm; Right: our designed ladybug leg with maximum tracking error of $\epsilon_{\max} = 4.1$ mm. (d) Left: input lamp shape; $\epsilon_{\max} = 32.9$ mm. Right: our designed lamp; $\epsilon_{\max} = 7.0$ mm. In all comparisons, we optimize cable forces and pulley positions.

we insert fabricable templates and directly optimize their parameters, using hard constraints to ensure fabricability. The fabricated characters are expressive, and closely match the simulation.

Our material optimization can be understood as varying stiffness properties at the microscopic scale, while our template optimization and search-and-replace strategy can be interpreted as optimization at the mesoscopic scale, to achieve an overall, desired macroscopic behavior.

In our system the user specifies the number of cables and the cable attachment points along the wire. For all our examples, few cables were needed to match targets with high accuracy, demonstrating the rich gamut of our spring-like entities. Optimizing the cable attachment position and the pulley placements on the rod would require parameterizations of the wire geometry. Putting users in control, they can select attachment points that avoid collisions with the wire and shell (see, e.g. the animatronic fish). We therefore shift the design complexity to the design of the wire shapes. However, an automated placement and sizing of a cable network could be beneficial.

As demonstrated in our Dinosaur and Alien Hand example, we capture the 3D visual appearance of the input model by attaching rigid and disconnected shells to our wire structure. It would also be interesting to model and attach a continuous and deformable shell to the wire structure.

Thanks to our optimization, all of our fabricated examples can match the input targets well. However, we occasionally observe small vibrations during some motions. In the future, we would like to investigate the dynamic properties of wires for inverse kinetic motion design. Intentionally exploiting wire-wire or wire-cable collisions for more expressive motions could be an interesting direction for future research.

Due to the wire being thin, and animated skeletons typically being collision-free, we have not seen problems with global collisions when designing our kinetic characters. However, if collisions were a problem one could add an additional objective to the design problem, penalizing non-local wire-wire proximity.

Due to the available consumer-grade hardware, we are restricted to automated fabrication of 2D and semi-automated fabrication of 3D shapes. Industrial-grade CNC benders such as the E-Flex by the BLM Group are capable of automatically fabricating 3D shapes, which would enable more complex and expressive 3D geometries. With our work, we show that locally planar templates can approximate a spatially-rich global deformation behavior. However, it would be interesting to explore the potential of 3D fabrication.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful comments; Maurizio Nitti and Alessia Marra for model and animation design; Christian Schüller and Jan Wezel for help with thermoforming the shell of our Ladybug; Zhao Ma for help with figure editing and video recording. This work has been supported by the SOMA project (European Commission, Horizon 2020 Framework Programme, H2020-ICT-645599).

REFERENCES

Moritz Bächer, Stelian Coros, and Bernhard Thomaszewski. 2015. LinkEdit: Interactive Linkage Editing Using Symbolic Kinematics. *ACM Trans. Graph.* 34, 4, Article 99 (July 2015), 8 pages.

Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete Viscous Threads. *ACM Trans. Graph.* 29, 4, Article 116 (July 2010), 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.

Amit H. Bermano, Thomas Funkhouser, and Szymon Rusinkiewicz. 2017. State of the Art in Methods and Representations for Fabrication-Aware Design. In *Eurographics State of the Art Reports.*

James M. Bern, Kai-Hung Chang, and Stelian Coros. 2017. Interactive Design of Animated Plushies. *ACM Trans. Graph.* 36, 4, Article 80 (July 2017), 11 pages.

D. Cellucci, R. MacCurdy, H. Lipson, and S. Risi. 2017. 1D Printing of Recyclable Robots. *IEEE Robotics and Automation Letters* 2, 4 (2017), 1964–1971.

Duygu Ceylan, Wilmot Li, Niloy J. Mitra, Maneesh Agrawala, and Mark Pauly. 2013. Designing and Fabricating Mechanical Automata from Mocap Sequences. *ACM Trans. Graph.* 32, 6, Article 186 (Nov. 2013), 11 pages.

Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W. Sumner, Wojciech Matusik, and Bernd Bickel. 2013. Computational Design of Mechanical Characters. *ACM Trans. Graph.* 32, 4, Article 83 (2013), 12 pages.

Akash Garg, Andrew O. Sageman-Furnas, Bailin Deng, Yonghao Yue, Eitan Grinspun, Mark Pauly, and Max Wardetzky. 2014. Wire Mesh Design. *ACM Trans. Graph.* 33, 4, Article 66 (July 2014), 12 pages.

Emmanuel Iarussi, Wilmot Li, and Adrien Bousseau. 2015. WrapIt: Computer-assisted Crafting of Wire Wrapped Jewelry. *ACM Trans. Graph.* 34, 6, Article 221 (Oct. 2015), 8 pages.

Robert K. Katzschmann, Joseph DelPreto, Robert MacCurdy, and Daniela Rus. 2018. Exploration of underwater life with an acoustically controlled soft robotic fish. *Science Robotics* 3, 16 (2018).

Martin Kilian, Aron Monszpart, and Niloy J. Mitra. 2017. String Actuated Curved Folded Surfaces. *ACM Trans. Graph.* 36, 3, Article 25 (May 2017), 13 pages.

Jian Li, Sheldon Andrews, Krisztian G. Birkas, and Paul G. Kry. 2017. Task-based Design of Cable-driven Articulated Mechanisms. In *ACM Symposium on Computational Fabrication (SCF '17)*. Article 6, 12 pages.

M. Lin, T. Shao, Y. Zheng, N. Mitra, and K. Zhou. 2017. Recovering Functional Mechanical Assemblies from Raw Scans. *IEEE Transactions on Visualization and Computer Graphics* PP, 99 (2017), 1–1.

Lingjie Liu, Duygu Ceylan, Cheng Lin, Wenping Wang, and Niloy J. Mitra. 2017. Image-based Reconstruction of Wire Art. *ACM Trans. Graph.* 36, 4, Article 63 (July 2017), 11 pages.

Vittorio Megaro, Espen Knoop, Andrew Spielberg, David I. W. Levin, Wojciech Matusik, Markus Gross, Bernhard Thomaszewski, and Moritz Bächer. 2017. Designing Cable-driven Actuation Networks for Kinematic Chains and Trees. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA '17)*. Article 15, 10 pages.

Vittorio Megaro, Bernhard Thomaszewski, Damien Gauge, Eitan Grinspun, Stelian Coros, and Markus Gross. 2014. ChaCra: An Interactive Design System for Rapid Character Crafting. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '14)*. Eurographics Association, 123–130.

Vittorio Megaro, Jonas Zehnder, Moritz Bächer, Stelian Coros, Markus Gross, and Bernhard Thomaszewski. 2017. A Computational Design Tool for Compliant Mechanisms. *ACM Trans. Graph.* 36, 4, Article 82 (July 2017), 12 pages.

Eder Miguel, Mathias Lepoutre, and Bernd Bickel. 2016. Computational Design of Stable Planar-rod Structures. *ACM Trans. Graph.* 35, 4, Article 86 (July 2016), 11 pages.

Niloy J. Mitra, Yong-Liang Yang, Dong-Ming Yan, Wilmot Li, and Maneesh Agrawala. 2010. Illustrating How Mechanical Assemblies Work. *ACM Trans. Graph.* 29, 4, Article 58 (July 2010), 12 pages.

Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization, second edition.* World Scientific.

Julian Panetta, Abtin Rahimian, and Denis Zorin. 2017. Worst-case Stress Relief for Microstructures. *ACM Trans. Graph.* 36, 4, Article 122 (July 2017), 16 pages.

Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. 2015. Elastic Textures for Additive Fabrication. *ACM Trans. Graph.* 34, 4, Article 135 (July 2015), 12 pages.

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

Christian Schüller, Daniele Panozzo, Anselm Grundhöfer, Henning Zimmer, Evgeni Sorkine, and Olga Sorkine-Hornung. 2016. Computational Thermoforming. *ACM Trans. Graph.* 35, 4, Article 43 (July 2016), 9 pages.

Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, and Markus Gross. 2015. Microstructures to Control Elasticity in 3D Printing. *ACM Trans. Graph.* 34, 4, Article 136 (July 2015), 13 pages.

Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross. 2013. Computational Design of Actuated Deformable Characters. *ACM Trans. Graph.* 32, 4, Article 82 (July 2013), 10 pages.

Peng Song, Xiaofei Wang, Xiao Tang, Chi-Wing Fu, Hongfei Xu, Ligang Liu, and Niloy J. Mitra. 2017. Computational Design of Wind-up Toys. *ACM Trans. Graph.* 36, 6, Article 238 (Nov. 2017), 13 pages.

Bernhard Thomaszewski, Stelian Coros, Damien Gauge, Vittorio Megaro, Eitan Grinspun, and Markus Gross. 2014. Computational Design of Linkage-based Characters. *ACM Trans. Graph.* 33, 4, Article 64 (July 2014), 9 pages.

Jonas Zehnder, Stelian Coros, and Bernhard Thomaszewski. 2016. Designing Structurally-sound Ornamental Curve Networks. *ACM Trans. Graph.* 35, 4, Article 99 (July 2016), 10 pages.

Ran Zhang, Thomas Auzinger, Duygu Ceylan, Wilmot Li, and Bernd Bickel. 2017. Functionality-aware Retargeting of Mechanisms to 3D Shapes. *ACM Trans. Graph.* 36, 4, Article 81 (July 2017), 13 pages.

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)*. 179–188.

Bo Zhu, Mélina Skouras, Desai Chen, and Wojciech Matusik. 2017. Two-Scale Topology Optimization with Microstructures. *ACM Trans. Graph.* 36, 5, Article 164 (July 2017), 16 pages.

Lifeng Zhu, Weiwei Xu, John Snyder, Yang Liu, Guoping Wang, and Baining Guo. 2012. Motion-guided Mechanical Toy Modeling. *ACM Trans. Graph.* 31, 6, Article 127 (Nov. 2012), 10 pages.