

Computational Design of Linkage-Based Characters

Bernhard Thomaszewski^{*1}

Stelian Coros^{*1}

Damien Gauge²

Vittorio Megaro²

Eitan Grinspun³

Markus Gross^{1,2}

¹Disney Research Zurich

²ETH Zurich

³Columbia University

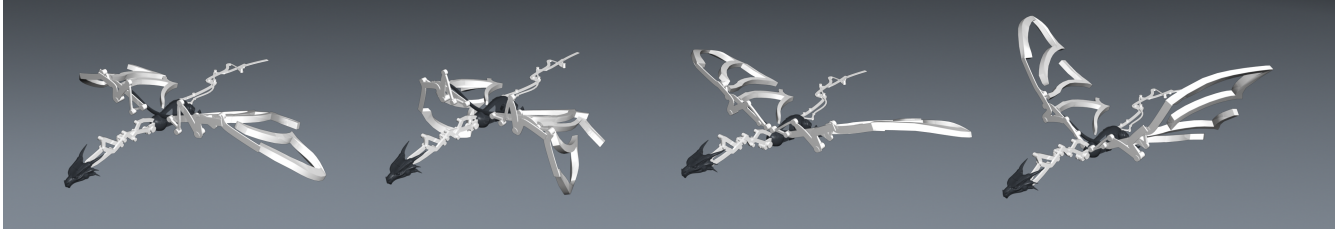


Figure 1: Our design system allows the user to create linkage-based characters that are able to perform complex and compelling motions.

Abstract

We present a design system for linkage-based characters, combining form and function in an aesthetically-pleasing manner. Linkage-based character design exhibits a mix of discrete and continuous problems, making for a highly unintuitive design space that is difficult to navigate without assistance. Our system significantly simplifies this task by allowing users to interactively browse different topology options, thus guiding the discrete set of choices that need to be made. A subsequent continuous optimization step improves motion quality and, crucially, safeguards against singularities. We demonstrate the flexibility of our method on a diverse set of character designs, and then realize our designs by physically fabricating prototypes.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: mechanical characters, linkage design, animation, fabrication, interactive design

Links:  DL  PDF

1 Introduction

From Heron’s *Pneumatics*, to da Vinci’s *Mechanical Lion* and von Kempelen’s *Automaton Chess Player*—animated mechanical characters have intrigued mankind for millennia [Bedini 1964]. Unquestionably, these examples are creations of ingenious designers, who were at the same time skillful engineers. With the advent of 3D printers, the machinery for *fabricating* personalized mechanical

toys, artifacts, or even robots becomes widely available. Without the assistance of specialized software, however, *designing* such animated characters is a virtually impossible task for casual users.

A standard approach to designing animated physical characters is to start from a skeletal base consisting of rigid bodies that are connected through joints. In order to control the motion of the character, an actuator such as an electric motor or a hydraulic element is placed at every joint. While this design allows for easily reconfigurable motions, the large number of actuators also entails problems in terms of cost, weight, and control. Additionally, even if the character exhibits a complex motion path involving all joints, this motion is typically confined to a low-dimensional subspace. Clearly, having an actuator for every joint is unnecessary if the character is performing a cyclic motion that could be controlled through a single active degree-of-freedom (DoF).

An alternative to using many motors is to resort to compound mechanisms that can perform intricate motions. Inspired by the work of contemporary artists such as Werner Gergaut [2011] and Andrew Chase [2014], we consider *mechanical linkages*—networks of rigid links and one degree-of-freedom joints. In engineering, linkages are employed for their functionality: to transform a given input force and movement into a desired output force and movement. However, in the hand of a skillful designer, the linkage mechanism itself can become a work of art. The shape of the rigid links, their motions and the overall silhouette of the mechanisms can lead to appealing kinematic sculptures or representations of animated characters; see, e.g., the linkages by Klann and Jansen or the kinetic sculptures by Gergaut.

Despite the fact that linkages are made from very basic components, they can create complex and often surprising motions. Indeed, even for seemingly simple linkages, it is often difficult to predict their motion. Not surprisingly, the inverse problem of determining a linkage that will reproduce or best approximate a desired motion is even harder: one has to determine its *topology*, i.e., how its components are connected, and its *parameters*, i.e., the dimensions of the components and their connection points.

Kinematic Linkage Design Finding a good or even optimal topology for a linkage structure is difficult. Indeed, the general problem of kinematic linkage design has fascinated mathematicians for centuries. Kempe’s 1876 Universality Theorem, for instance, states that planar curves defined by polynomial equations can be traced out by linkages, leading to the famous “there is a

^{*} The first two authors contributed equally.

linkage that signs your name” quote [King 1998]. The proof of the theorem relies on a few elementary types of linkages, reversors, additors, translators and multipliers, that are connected to each other to create complex linkage systems. Unfortunately, the theory also predicts that the number of such elementary linkages that must be composed together quickly becomes intractable in practice. Others have studied simpler linkage systems with fixed topology, which can be used to reproduce certain types of motions. Burmester’s circlepoint-centerpoint theory [1888] and Freudenstein’s equations [1954], for instance, can be used to compute the parameters of four-bar linkages such that one of the links moves through several prescribed configurations. These approaches rely on complex algebraic formulas, and the need for interactive programs that aid in the design process have been recognized since the early days of computers [Kaufman and Maurer 1971].

Overview & Contributions We seek a design tool that enables non-experts to create linkage-based characters—characters that integrate form and function in a seamless, aesthetically-pleasing manner. Finding good linkage topologies and geometry becomes even more difficult when such aesthetic requirements must be considered. Therefore, to find simple and elegant linkage mechanisms that account for these subjective criteria, we opt for a system that puts the user in the design loop.

The design process begins from a given, fully-actuated mechanical structure performing some desired periodic motion. This input data is represented by a set of rigid members (*links*), interconnected by joints, with periodic time-varying angles prescribed for each joint. We consider the case where each motion complex (legs, arms, tails) acts within a plane, but the skeleton may include multiple complexes exhibiting motions on multiple planes.

Our design tool enables the user to successively replace the joint motors with new rigid links that mechanically couple the motions of different parts of the mechanical assembly. Manually performing this operation can be a very tedious and frustrating task. The space of possible designs is large, highly nonlinear, and presents both discrete and continuous parameters. This combination of challenges leads to frequent design pitfalls including poor reproduction of the desired motion, or worse, mechanical *locking* of the assembly, barring motion altogether. Our computational design system circumvents these difficulties by allowing the user to quickly explore multiple valid configurations for the intermediate linkage structures. In order to make this possible, we (a) develop a fast method to compute approximately optimal parameters for new mechanical linkages, (b) provide an analysis of singularities that can arise in linkage systems, and (c) propose a method to optimize the motion of linkage-based characters and their robustness against fabrication errors. The design process thus becomes tractable and enjoyable; as indicated by our results and experiments, even non-expert users can quickly create aesthetically-pleasing animated characters.

2 Related Work

Video games and CG movies captivate audiences world-wide because they place no limits on the types of environments, characters and objects that can be created. Rapid manufacturing devices hold the promise of bringing this type of freedom to the real world by allowing anyone to create their own personalized, physical objects. The task of designing objects that can be fabricated, however, is typically very challenging and therefore beyond the capabilities of casual users. Computational tools that simplify the design process are therefore becoming increasingly important—and actively pursued within graphics research. It is now possible, for instance, to automate the design of manufactured objects with respect to appear-

ance properties [Weyrich et al. 2009; Hasan et al. 2010; Dong et al. 2010], structural stability [Stava et al. 2012; Zhou et al. 2013; Wang et al. 2013], or deformation behavior [Bickel et al. 2010; Bickel et al. 2012; Skouras et al. 2012]. A variety of interactive design systems have been proposed to allow casual users to design and customize different types of objects, such as plush toys [Mori and Igarashi 2007], furniture [Lau et al. 2011; Umetani et al. 2012], or beadwork [Igarashi et al. 2012].

Particularly challenging is the task of creating physical representations of virtual characters. This is an emerging research area and several recent works have started to investigate its various aspects. For instance, Bäcker et al. [2012] and Cali et al. [2012] proposed methods to automate the design of 3D printable representations of digital characters with functional joints. Prévost and colleagues [2013] introduced a method that alters the internal material distribution of 3D printed characters to allow them to balance while standing. Skouras et al. [2013] also present a method to automatically optimize the internal distribution of material parameters, but with the goal of controlling the ways in which 3D printed objects deform when subjected to external forces.

Our work builds on recent efforts to transform virtual characters into animated, tangible artifacts [Zhu et al. 2012; Coros et al. 2013; Ceylan et al. 2013]. The method of Zhu et al. [2012] uses mechanical devices such as cams and followers or crank-sliders to create animated toys. The method proposed by Ceylan et al. [2013] creates specialized mechanical oscillators that are placed at every joint of the character, and linked to each other through belts and pulleys. The system proposed by Coros et al. [2013] uses parameterized mechanisms with fixed topology in order to drive the motion of a mechanical character’s end effectors. Each one of these design tools employs a library of parameterized mechanical assembly templates with fixed topology.

Our goal is set apart from these three works by our interest in facilitating the design of original assemblies with arbitrary geometry and topology. In doing so, we enable the coupled, simultaneous design of both character shape and driving mechanism. This makes it possible to design *elegant* mechanisms that form a part of the designed artifact, rather than a hidden internal; we enable the design of structures whose components fulfill simultaneous requirements on both form and function.

3 Kinematics

Our system automates the process of designing linkage-based mechanical characters that are capable of compelling movements. The main design phase operates on planar linkages that represent different parts of the character. In a post process, we apply global transformations to the individual linkages output by our framework in order to create full-body mechanical characters whose motions can be either planar or in 3D. For the remainder of the paper, we therefore restrict our discussion to planar mechanisms.

We model linkage-based characters as sets of rigid components that are interconnected through constraints. Each component c_i has three degrees of freedom $\mathbf{s}_i = (\alpha, x, y)^T$ describing its orientation and position in world-space. We assemble the degrees of freedom of all components into a global state vector $\mathbf{s}$. As is common in rigid body kinematics, we introduce notation to refer to the world-space coordinates of a specified rigid body material point. Let $\mathbf{q}_j = (u, v)$ be the local coordinates of a material point on rigid component c_i . The world position of this point is given by the nonlinear transformation $\mathbf{x}(\mathbf{q}_j, \mathbf{s}_i) = \mathbf{R}_i \mathbf{q}_j + \mathbf{t}_i$, where $\mathbf{R}_i$ is the rotation matrix corresponding to the angle α and $\mathbf{t}_i$ is the translation corresponding to position variables $(x, y)^T$. We use $\mathbf{x}_i^j(t)$ as a

shorthand for $\mathbf{x}(\mathbf{q}_j, \mathbf{s}_i(t))$ at a specific time instance denoted by t .

The rigid components are connected to each other through constraints $\{\mathbf{C}_i\}$ that restrict their relative motion. We use three types of constraints for this work. Pin joint constraints are defined by the local coordinates of the pin, $\mathbf{q}_{p1}$ and $\mathbf{q}_{p2}$, on the two links l_1 and l_2 that are connected to each other. The constraints take the form $\mathbf{C}_i(t) = \mathbf{x}_{l_1}^{p1}(t) - \mathbf{x}_{l_2}^{p2}(t)$ and ensure that the pin locations on the two bodies coincide in world coordinates. Motor constraints are used to specify the relative orientation between two links as a function of time: $C_i(t) = (\alpha_{l1}(t) - \alpha_{l2}(t)) - m(t)$. Here, $m(t)$ denotes the time varying angles output by the motor connecting the two components. Finally, we can explicitly fix some components in world-space to model the support structure of the character. For this purpose, we use constraints that set fixed target values for all the state variables of a component.

To step the assembly forward in time, we first update the motor angles and collect the individual constraints into a vector $\mathbf{C}$. We then determine the new state of the assembly by minimizing the constraint penalty

$$E_c(\mathbf{s}) = \frac{1}{2} \mathbf{C}(\mathbf{s})^T \mathbf{C}(\mathbf{s}). \quad (1)$$

The Hessian of this energy is sparse, symmetric and positive semi-definite, traits that specialized linear solvers can leverage for speed.

4 Design Loop

The input to our system consists of a virtual character performing a periodic motion which is specified as a set of time-varying angles for each joint of its skeletal structure. We begin by creating a motorized assembly that exactly matches the input motion. For this purpose, we transform each bone of the input skeleton into a rigid link, each joint to a pin constraint, and we create virtual motors that reproduce the joint trajectories as specified in the input animation. We create one additional component that is connected to a constant-speed *phase driver*, corresponding to the manually-operated crank or actuator that will drive the motion of the mechanical character. The goal of our system is to design a linkage structure, actuated by this phase driver, whose motion matches the motorized assembly as closely as possible.

Our design tool consists of two distinct phases. First, the motors driving the joints of the initial assembly are iteratively replaced with new rigid links that mechanically couple different parts of the assembly. The goal of this step is to define the topology of the linkage system, and to provide the global optimization stage with a functional mechanism. In the second phase, all the parameters of the mechanism are concurrently optimized in order to obtain an assembly whose motion is as close as possible to the input motion. This optimization step also ensures that the resulting linkage structures always remain a safe distance away from singularities, which serves two important purposes: first, it helps to lower the forces needed to actuate the resulting assemblies; second, it ensures that the assemblies can still function properly when mechanical play is inevitably added during the manufacturing process.

4.1 Linkage Topology Design

In order to create linkage systems that are as simple as possible, we follow one simple rule: *remove one motor, add a link*. It can be easily verified that this operation preserves the invariant that there are always as many constraints as degrees of freedom in the linkage system: removing a motor eliminates one constraint, adding the link introduces three DOFs, and two pin joints used to connect the new link to the existing structure result in four additional constraints.

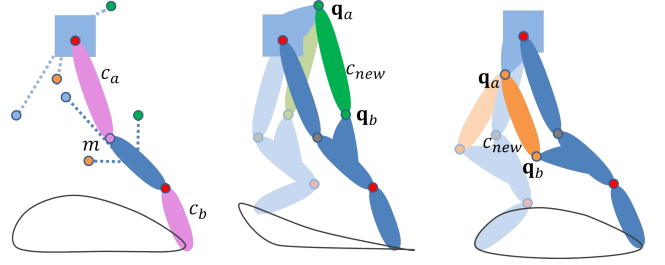


Figure 2: A motor (m) is being replaced by connecting components c_a and c_b through a new link (c_{new}). Left: our system generates a set of point pairs with locally-minimal distance variance. Different motions result when inserting a link between the pair with largest (middle) or smallest variance (right).

The motion of each component is thus either directly driven by a motor, or is mechanically coupled to that of other components. The linkage systems resulting from these elementary operations are minimal in the sense that this invariant cannot be maintained with fewer additional components.

The intuition for our approach is based on the following observation: if the distance d between two points on a pair of components does not change over an entire motion cycle, then these components can be connected through pin joints to a new rigid link of length d . Although the resulting mechanism would technically be over-constrained, the new link and its pin joints would be completely redundant. Removing a motor along the kinematic chain between the two components would resolve this redundancy while perfectly preserving the original motion of the linkage. In general, there are no guarantees that such pairs of points always exist. However, we observe that if the imposition of a fixed-distance constraint on point pair $\mathbf{q}_a$ and $\mathbf{q}_b$ induces minimal disturbance on the motion of the entire assembly, then in particular it induces minimal disturbance to the world-space distance $\|\mathbf{x}_a(t) - \mathbf{x}_b(t)\|$. Using the converse of this statement as intuition, we approximate the desired optimization by seeking a pair of points with *minimal distance variation*.

Distance Variance Minimization Given two components, c_a and c_b , we seek to find a pair of material points $\mathbf{q}_a$ and $\mathbf{q}_b$ whose world-space distance varies least throughout the motion (see Fig. 2, left). The mean squared world-space distance between these two points is given by

$$l_{ab} = \frac{1}{n_s} \sum_i^{n_s} \|\mathbf{x}_a(t_i) - \mathbf{x}_b(t_i)\|^2, \quad (2)$$

where n_s is the number of discrete time samples that span the entire motion cycle. The variance of this quantity is

$$\delta_{ab} = \frac{1}{n_s} \sum_i^{n_s} (\|\mathbf{x}_a(t_i) - \mathbf{x}_b(t_i)\|^2 - l_{ab})^2. \quad (3)$$

Since first and second derivatives of this expression are readily computed, (3) can be minimized efficiently using Newton’s method in order to obtain a pair of points, $\mathbf{q}_a$ and $\mathbf{q}_b$, with locally-optimal distance variance. Eq. (3) is fourth-order in the parameters we seek and will generally have multiple local minima (see Fig. 2), but not all of them are equally desirable. Fig. 3 shows an example in which the two components that are to be connected already share a common link. This existing link does not change its length, such that its two pin joint locations constitute a global minimum to (3). However, this trivial solution is clearly not helpful. In certain scenarios,

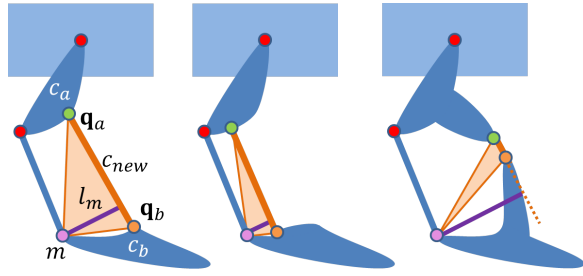


Figure 3: A motor (m) is selected and replaced with a new link (c_{new}) that connects components c_a and c_b . Small moment arms (middle) and short new links (right) are avoided by preventing the area of triangle (q_b, m, q_a) from becoming too small.

it is straightforward to identify such unwanted solutions, discard them, and restart the optimization using a different initial guess. However, we found that this trivial solution is often chosen by the optimization and, as the complexity of the linkage system increases, it becomes more and more difficult to detect such problems using heuristics. We address this issue by changing the landscape of the optimization problem.

As shown in Fig. 3, when the selected motor (m) is removed, the new link (c_{new}) becomes responsible for driving the motion of component c_b by directly coupling it to component c_a . The force transmitted through c_{new} is transformed into a torque acting on the joint where the motor was removed, causing component c_b to rotate about it. The torque is proportional to the length of the moment arm l_m . It can be seen from Fig. 3 that, as the coordinates of the pin joints move closer to the joints of the intermediate link c_m (the global minimum of (3) in this scenario), the moment arm decreases and the force needed to generate the same torque increases at an inversely proportional rate. In order to avoid the need for such high forces, we must ensure that the moment arm remains sufficiently large for all assembly states in the original motion. We found that optimizing the moment arm directly often leads to the new link decreasing in size while moving further away from the assembly (Fig. 3, right). To discourage this undesirable effect, we instead optimize for the area of the shaded triangle, which accounts for both the length of the new link and the moment arm. More formally, we ask that this area be at a safe distance from zero, using a penalty term that grows exponentially with decreasing area as

$$E_{\text{Area}} = -\log \sum_i^{n_s} \text{area}(\mathbf{x}_b(t_i), \mathbf{x}_m(t_i), \mathbf{x}_a(t_i))^2, \quad (4)$$

where $\mathbf{x}_m(t_i)$ denotes the world-space position of the motor m . We then minimize a weighted combination of (3) and (4) to compute the length of the new link and the locations of the pin joints, q_a and q_b . We weigh the variance term (3) ten times higher than the area term (4) for all our experiments. It is worth noting that unless the distance variance between the pair of selected points is zero, inserting the new link using the optimized parameters will change the resulting motion. Consequently, because the moment arm objectives are evaluated at assembly configurations obtained when the joint motors are still active, it is possible that the modified assembly will not function properly (see Sec. 5.1). Fortunately, such undesirable motor replacements are easily detected by simulating the modified assembly through a full cycle. If required, we randomly select a new initial guess for the parameters and restart the optimization process.

User-Guided Topology Design Using the optimization method described above, we compute candidate pairs of points where new

links can be inserted as motors are iteratively removed. As mentioned before, not all of these candidates are equally suitable. Since the distance variation metric does not explicitly measure the global impact of the motor replacement operations, it is possible that a change in motion due to a newly inserted link is not tolerable for the user. But even if the resulting motion is satisfying, it is still possible that the new link does not comply with the original aesthetic intent of the user. For these reasons, we allow the user to interactively direct the topology design process.

The user first selects a motor (m) to be replaced, together with a pair of components c_a and c_b , as illustrated in Fig. 2. The variance optimization step is then initiated using different initial parameters for q_a and q_b . Each optimization is very efficient, taking less than 0.1s to converge. As valid solutions are found (Fig. 2, middle and right), they are presented to the user who can pre-visualize the effect of different replacement choices (typically 3 to 6) on the motion and aesthetics of the assembly. Once a satisfying solution is found, the choice is committed and the operation is finalized. If none of the resulting options are acceptable, the user can repeat the process by selecting a different motor or a different pair of components to connect. The process terminates once all joint motors have been replaced and at least one of the links is connected to the input driver, which is treated as any other component in the assembly.

Adding Auxiliary Links Our system allows its users to quickly explore different topology options for the assembly under design. The ability to pre-visualize the impact of design choices is indeed very important, since even simple input animations already exhibit a multitude of possibilities for motor replacements (see Fig. 2). This diversity of choices allows for ample creative design space, which the user can navigate efficiently using our system.

In order to further personalize the appearance of the character, we allow the user to add additional, auxiliary structures to the linkage system. As illustrated in Fig. 8, right, these structures consist of two connected links that the user attaches to two existing components. Auxiliary structures do not affect the functionality of the mechanism, as their motions are completely specified by the components to which they are attached. But while their role is primarily aesthetic, they further extend the range of possible designs for the linkage systems. For instance, for our *Taurus* character (Fig. 8), auxiliary structures were used in the design of both the front and hind legs to delineate their contours.

5 Global Optimization

The output of the topology design stage consists of functional linkage systems. In order for the user to guide the design process, we implemented an efficient method that allows multiple options to be quickly and intuitively explored. While the distance variance minimization allows us to quickly generate a variety of different design options, this criterion does not directly reflect changes in the motion of the assembly and, as a result, there is quite likely room for improvement. We therefore perform an additional, global optimization step, whose purpose is to fine tune the linkage structure in order to best match the input motion. The parameters we optimize for consist of the local coordinates of all the pin joints that were added during the topology design stage. Optionally, the locations of the pin joints for the auxiliary links can be omitted from the optimization if their position is important for aesthetic reasons.

In principle, the implicit function theorem can be used to compute gradients that measure how a given change in parameters affects the motion of the entire assembly [Coros et al. 2013]. However, we found that the solutions obtained using this method can often result in near-singular configurations, arguably due to the unstruc-

tured and high-dimensional nature of the linkage systems we aim to optimize. Such configurations must be avoided for two reasons.

First, such ill-conditioned assemblies are overly susceptible to mechanical play and other imprecisions introduced during manufacturing, leading to bifurcations in configuration space (see Fig. 4) or barring motion altogether. Second, near-singular configurations typically lead to a significant amount of the force applied through the input driver working directly against the mechanical structure of the linkage (see Sec. 4), thus increasing the likelihood of mechanical failure. While the moment arm objective expressed in (4) can avoid some types of singularities, it is neither necessary, nor sufficient for detecting singular configurations. We must therefore employ a more robust and general way of quantifying how well-conditioned our linkage systems are.

5.1 Avoiding Singular Configurations

Among the most challenging technical aspects that we must address is the question of the well-posedness of the system of constraints specified by the linkage structures. In this section, we identify the types of problems that must be quantified and avoided during the optimization process.

Since our mechanisms are governed by nonlinear equations, there are generally multiple configurations that satisfy all the constraints. Fig. 4 shows a simple example that allows for geometric interpretation. Assume that the motor (m) has been removed, and the pin joint parameters $\mathbf{q}_a$ and $\mathbf{q}_b$ are set. If the new link was not yet attached to c_b , then the set of valid world coordinates for the pin joint $\mathbf{q}_b$, as viewed from the coordinate frame of the new link, are given by the green circle. Similarly, the set of possible world coordinates for $\mathbf{q}_b$, as seen from the coordinate frame of c_b , are given by the gray circle, since the removed motor no longer controls the angle of the joint. The set of valid linkage configurations where all constraints are satisfied, once the pin connection between the new link and c_b is added, is given by the intersection between the two circles.

Two observations can be made regarding Fig. 4, *left*: first, the valid configurations are well-separated in state-space; second, the constraint gradients, i.e., the normals of the two circles (*red*) at the intersection points, are almost orthogonal. These two aspects make for a highly stable configuration that is robust against perturbations such as numerical errors in simulation or inaccuracies in the fabrication process. As the length of the new link is progressively changed, the intersection points approach each other and the constraint gradients begin to align (Fig. 4, *middle*). Eventually, the intersection points coincide and the constraint gradients become collinear (Fig. 4, *right*). Once such a singular configuration is reached, the motion of the assembly is no longer well defined, and the exact same inputs from the active actuators could cause link c_b to rotate either clock-wise or counter-clockwise. We refer to this problem as a *bifurcation* in the configuration space of the linkage.

At first thought, it is tempting to apply the moment arm objective (4) in order to prevent three consecutive joints from ever becoming collinear. However, collinearity is neither a necessary, nor a sufficient criterion for detecting the ill-conditioned assembly configurations. The inset figure shows an example of an ill-conditioned linkage for which no configuration leads to three collinear pin joints. In this scenario, a link (*purple*) has already been introduced in a previous step of the topology design stage. Following the argumentation from the previous paragraph, once the

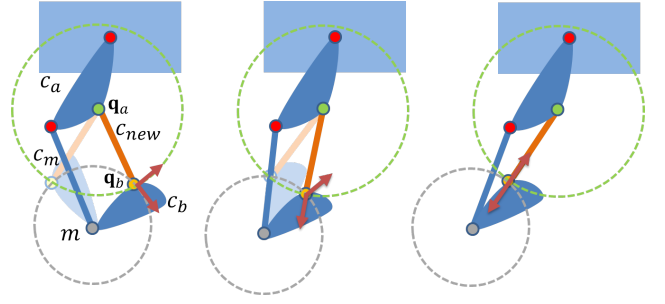
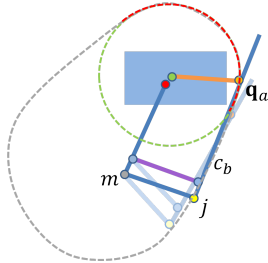


Figure 4: Replacing a motor (m) with a new link (c_{new}). The intersection points of the two circles correspond to valid configurations for $\mathbf{q}_b$. Well-separated intersections indicate good stability (left) that decreases as the points approach each other (middle). A bifurcation point occurs once the intersection points coincide (right).

the motor (m) is removed, the set of possible world coordinates for $\mathbf{q}_b$, as seen from the coordinate frame of c_b , are no longer given by a circle, but by a more complex shape, which is due to the structural coupling between the joints m and j . As a result, there is an entire subspace (indicated by the *red* arc) that satisfies all the constraints in the system and, consequently, the configuration of the linkage is not uniquely defined. It is worth noting that the constraint gradients become collinear, leading to an under-constrained system.

The previous example illustrates that the collinearity criterion is not sufficient for detecting ill-conditioned linkages. As shown in Fig. 5, the collinearity of three joints does also not necessarily imply that a singular configuration is encountered. Given the same kinematic structure, the placement of the motor (*red*) can either lead to an ill-defined (*left*) or well-defined (*right*) assembly.

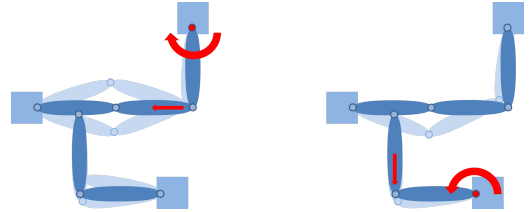


Figure 5: A counter-example demonstrating that collinearity is not sufficient for singularity. Depending on where the motor sits (*red*), a singularity can arise (*left*) or not (*right*).

Considering the complexity of these examples, we found it difficult to identify a simple set of heuristics that correctly avoid all types of problematic scenarios, motivating us to seek a more principled approach.

Singular Value Analysis As a generalization of the example shown in Fig. 4, we can draw conclusions on the stability of an assembly in a given configuration by analyzing the constraint Jacobian $\mathbf{J} = \frac{\partial \mathbf{C}}{\partial \mathbf{s}}$. As an important difference, it is not sufficient to check constraint gradients for pairwise collinearity. Instead, we have to ensure that no constraint gradient is in the span of (sets of) other gradients. Since, by construction, our mechanisms always have exactly as many constraints as degrees of freedom, this condition amounts to the Jacobian having full rank. If this is true at all times, the motion of the assembly will be well-defined.

Our goal is to obtain designs that are well posed in the sense described above. While there are faster ways to determine whether $\mathbf{J}$ has full rank, a singular value decomposition (SVD) provides valu-

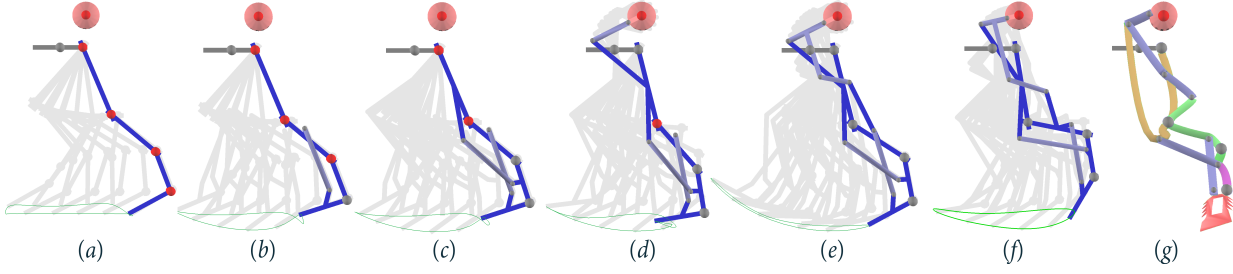


Figure 6: The character design process illustrated on the Satyr's leg: input animation (a), interactive motor replacement (b-e), after global optimization (f), and after linkage shaping (g).

able additional information. For non-degenerate cases, the magnitude of the smallest singular value provides a continuous measure of how far from singular a given configuration is. We can thus construct a penalty term that avoids singular configurations during the global parameter optimization stage, as described next.

5.2 Global Parameter Optimization

Once the topology has been determined, we run a global optimization process in order to improve the character with respect to its motion and its resilience to singularities. The latter point is particularly important since tolerances in fabrication can make a just-about-stable assembly topple. We therefore rely on the fail-safe singular value decomposition to detect degeneracies, but since its derivatives are not easily available, we opt for a derivative-free optimization scheme. We define a number of objectives and penalty terms that we use for this optimization.

Objectives We would like the character's shape and motion to stay as close to the input as possible, which we quantify with two objectives. The first one measures trajectory deviations as

$$E_{\text{marker}} = \sum_i^{n_m} \sum_j^{n_s} \frac{1}{2} \|\mathbf{m}_i(t_j) - \tilde{\mathbf{m}}_i(t_j)\|^2, \quad (5)$$

where $\mathbf{m}_i(t_j)$ and $\tilde{\mathbf{m}}_i(t_j)$ are sample positions for n_m marker points in the current configuration and input motion, respectively. Typically, a marker point is placed at every end effector, but other choices are possible as well. The second objective measures deviations in component states as

$$E_{\text{state}} = \sum_i^{n_m} \sum_j^{n_c} \frac{1}{2} \|\mathbf{s}_i(t_j) - \tilde{\mathbf{s}}_i(t_j)\|^2, \quad (6)$$

where n_c denotes the number of components and $\tilde{\mathbf{s}}(t_j)$ are the time-varying component states of the input motion. Furthermore, for links that were added during the design process, we ask that the positions of the corresponding pin joints be close to the original skeleton. To this end, we simply compute the distance between each of these $n + j$ joints $\mathbf{j}_i$ and the line segment $\mathbf{l}_i$ corresponding to the original component to which it connects,

$$E_{\text{joint}} = \sum_i^{n_j} \frac{1}{2} \text{dist}(\mathbf{j}_i, \mathbf{l}_i)^2. \quad (7)$$

Finally, in order to prevent degeneracies, we introduce a term that penalizes small singular values of the constraint Jacobian,

$$E_{\text{singular}} = \sum_j^{n_s} (\lambda_{\min}(t_j) + \varepsilon)^{-\alpha}, \quad (8)$$

where $\lambda_{\min}$ denotes the smallest singular value of the assembly, ε is a small positive threshold to avoid divisions by zero, and the exponent $\alpha > 0$ can be set to modulate how fast the penalty increases as $\lambda_{\min}$ goes to zero (typically, we set $\varepsilon = 10^{-8}$ and $\alpha = 2.0$).

A weighted sum of the energy terms introduced in (5)-(8) is used to define a global objective, which is minimized using a stochastic optimization scheme based on covariance matrix adaptation (CMA). The weights associated with the different objectives are set to 100, 100, 500 and 1 for E_{marker} , E_{state} , E_{joint} and E_{singular} respectively. We used the same weights for all of our experiments.

5.3 Character Finishing

While it is difficult to evoke an organic impression with a small number of straight bars, a few curved links can already create very expressive characters (cf. Fig. 6 f and g). To this end, we provide a simple spline tool that can be used to shape the links in the assembly as desired. Our system automatically computes the parameters of the spline segments such as to minimize the change in tangent directions between adjacent links over the entire motion. The magnitudes of the resulting tangents modulate the tension of the splines, and they can be adjusted by the user in order to stylize the mechanical characters.

Our system does not check for collisions between different components of the linkage structures during the design stage. In order to prevent self-intersections, we therefore offset the rigid links along the direction normal to their motion plane. This layering operation is performed in a post-processing step. Although we found it sufficiently easy to manually assign the components to different depth layers for the examples we demonstrate, the automated layering approach of [Coros et al. 2013] could be employed as well. Once all the components move without collisions, we procedurally perform the boolean operations required to create the geometry of the pin joints, after which the character is ready for fabrication.

6 Results

We used our system to design eight linkages, forming four different characters. We further validated one design by fabricating a physical prototype. In the following we discuss our results and comment on our experience with the proposed interactive design tool.

6.1 Examples

Overview Fig. 6 shows a summary of the design process for one leg of our *Satyr* character, a creature from Greek mythology with goat-like features. Starting from an input animation of the Satyr's leg (a), the user progressively removes motors (shown as red dots)

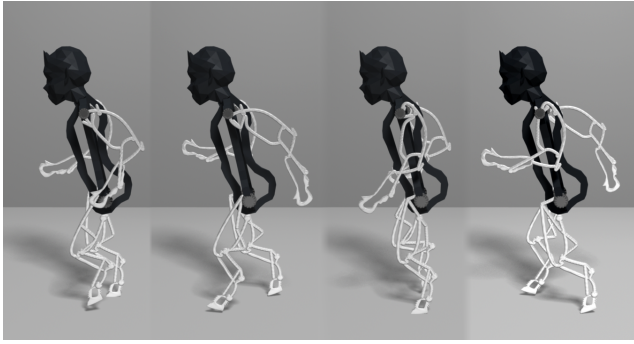


Figure 7: Snapshots from the *Satyr*'s motion cycle.

and mechanically couples different parts of the assembly using additional rigid links (*b-e*). At each step in the process, the user is provided with a set of preview animations illustrating different motor replacement operations that would lead to valid assemblies. Having decided on one of the options, the new link is added to the assembly, the motor is removed and the process repeats. Once all the joint motors have been replaced (*e*), the global optimization method is invoked in order to improve the assembly with respect to motion quality, robustness to singularities and other criteria as described in Sec. 5.2. One may note that after the global optimization step (*f*), the resulting motion is closer to the input and that the mechanical links inserted during topology design moved closer to the assembly, leading to a more compact visual appearance. Finally, as a post-process, the user is provided with intuitive tools to alter the shape of the links while maintaining their functionality (*g*). A complete account of this design session can be found in the accompanying video. The finished design of the character can be seen in Fig. 7.

Auxiliary Links and Shaping Fig. 8 shows *Taurus*, a fiercely charging bull in full gallop. This example is inspired by the *Mechanisches Pferd* model [Gergaut 2011], and it showcases the ability of our method to automate the design of mechanical characters whose complexity is similar to those built by human experts. Each leg was designed starting from an input skeleton with four joint motors. During the first stage of our design system, the legs were augmented with an auxiliary pair of links (Fig. 8, *right*). While serving a primarily aesthetic purpose, the auxiliary links can also be used during the motor replacement operations. The user can furthermore specify whether the kinematic parameters of the auxiliary links should be considered during global optimization, in which case their exact initial placement is not critical.

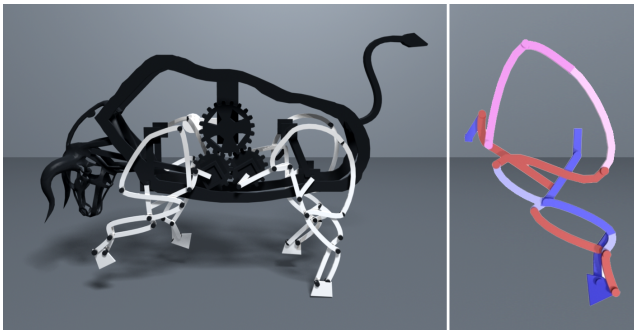


Figure 8: Left: the *Taurus* character. Right: the original leg structure (blue) was augmented with mechanical coupling (red) and auxiliary links (pink) during topology design. All links were reshaped in a post process.

The *Taurus* example also demonstrates how linkage re-shaping can support and reinforce the character's personality in a stylized and organic way. It is worth noting that the only parameter that needs to be set for this purpose is the global tension of the splines, the tangent directions are set automatically in order to obtain smooth segment transitions for the entire motion.

Globally-Coupled Motion All examples discussed thus far exhibit closed kinematic loops comprising ten or more interconnected links. When we first began our investigations, we considered two alternate strategies to pursue: (a) the development of an automated construction algorithm based on *composition of basic linkages*, e.g., based on Kempe's 1876 Universality Theorem; and (b) geometric optimization of a single linkage drawn from a *library* of precomputed templates, essentially forgoing linkage *design* and focusing instead on *reuse*. Having gained experience with our proposed approach, we believe it demonstrates advantages over both alternatives. On the one hand, for a given target motion, our linkages are much more compact, and elegant, than what would be possible by composing basic linkages. On the other hand, while a precomputed *library* may be feasible for four and five bar linkages, the space of desirable linkages with ten or more bars is exceedingly large, making both the creation of the library, and the retrieval task, a major (if not intractable) undertaking. Most of all, however, we find it impossible to discount the insights and inspiration that occur when the artistic and engineering design processes are intimately interwoven.

While it is often possible to design different parts of a character in isolation (arms, legs, etc.), many characters exhibit globally-coupled motions that cannot be easily separated into individual parts, and thus must be designed as an ensemble. *Lanternino*, a stylized dancing lantern, is an example of such a character: bendy knees, a curvy spine and rhythmically swinging arms and head make for a highly coupled motion. Despite the complexity of the input motion for this character and the size of the discrete design space, creating its topology took about 20 minutes with our system. As with all of our examples, there is only a single input motor in the final design of this character. The motion of the input motor is transmitted to the linkage system using three rotating components that are attached to the body of the character. One motor and a gear train would therefore be sufficient to actuate this character.

Motion in Different Planes Our system allows the user to create mechanical characters made from complex planar linkages. However, it is possible to combine linkages moving in different planes in order to create compelling characters featuring three-dimensional motions. For our *Dragon* character, for example, the neck, wings, and tail linkages move in three orthogonal planes (see Fig. 1). As another noteworthy aspect of this example, in order to allow for the

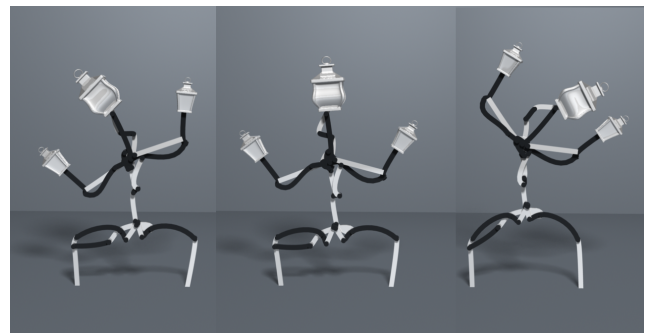


Figure 9: *Lanternino*: a stylized dancing lantern.

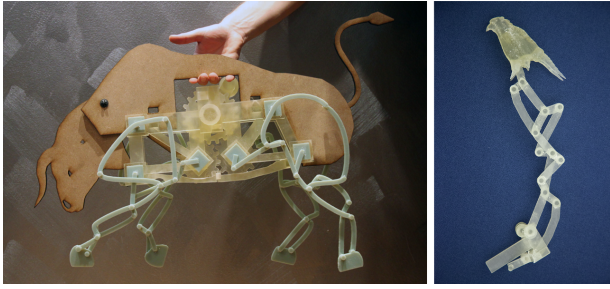


Figure 10: Fabricated prototypes showcasing the four linkage systems of the Taurus character (left) and the neck of the Dragon character (right).

head to stay level with the main body throughout the motion cycle, its joints have to move in unison as they undergo large relative rotations. The global optimization step was able to find parameters for the linkage system that reproduce this important aspect of the character’s motion. We emphasize that, while it was easy for the artist to produce this motion using keyframe animation, creating the mechanical coupling without assistance from the computer is intractable. Our system, however, allows the user to straddle the gap between computer-generated animation and its physical realization.

6.2 Experience & Validation

Using the Interface We ran an extensive set of experiments with our interface and report on some high-level findings here. The interactive motor replacement operation is a central stage of our pipeline in which the user lays out the topology of the assembly. In general, there are multiple options for each motor replacement step, leading to a combinatorial number of possible linkage topologies for the final assemblies. While it is impractical for the user to explore the entire space of possibilities, we found that, using our system, distilling out the most relevant options becomes a manageable task. This is because a large number of topologies can be pruned at each step. For instance, if a connection between two links negatively affects the overall look of the mechanical characters, the user will likely choose not to explore that branch of possible mechanism designs.

For the same final topology, the order in which motors are replaced can also affect the resulting motion. After the global parameter optimization stage, however, these differences largely diminish. For different final topologies, we notice that motions can differ even after optimization. However, we were able to find assemblies that perceptually match the input motion for all the experiments we have performed. In addition, the variations are typically small enough to warrant ample room for artistic freedom in the topology design.

	# Actuators	# Comp. (input)	# Comp. (final)
Taurus’ leg	4	5	11
Satyr’s leg	4	5	9
Satyr’s arm	2	3	5
Dragon’s wing	3	4	7
Dragon’s tail	4	5	9
Dragon’s neck	5	6	11
Lanternino	9	12	21

Table 1: An overview of the complexity of our example characters and their individual linkage systems.

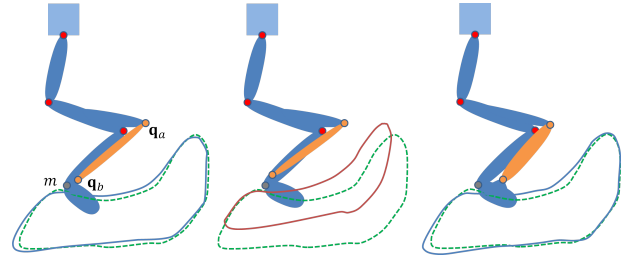


Figure 11: Using only motion objectives for global optimization (left), the input motion (green) is well approximated, but small parameter perturbations cause a state flip in the hoof (middle). Using our SVD objective (right) the assembly is robust against perturbations.

Robustness Preventing singularities during design is crucial in order to obtain well-functioning characters that are robust against fabrication errors and wear. We use the area term (4) described in Sec. 4 to avoid many problematic cases during the topology design step, as it is computationally very efficient and enables interactive browsing of different possible options. For the global optimization step, however, this criterion is not sufficient to safeguard against singularities. Fig. 11 shows an example that illustrates the difference between using only motion quality objectives during global optimization (left) and additionally using the non-singularity term described in Sec. 5.1 (right). In the first case, the optimization finds a parameter set that leads to good motion approximation, but brings the assembly very close to a singularity at one point during the motion cycle. The fragility of this design is illustrated in Fig. 11, middle. To mimic the effect of mechanical play, we added a small perturbation to the parameters of the assembly, leading to the hoof flipping to a different stable, yet unnatural, configuration. By contrast, our SVD-based criterion safeguards against such ill-conditioned designs, leading to an assembly that is robust to the same perturbation while still achieving high motion quality (right).

Performance Table 1 provides an overview of the complexity of all examples presented in this section. The time spent on topology design was below half an hour in all cases. However, only a small fraction of this time is taken up by distance variance minimization, whereas during most of the time, the user is exploring different design options. For the Satyr’s leg, for instance, it took about 0.1s on average to solve the minimization problem, and less than 10s to compute six valid and visually distinct solutions. Furthermore, the number of minima found for the different motor replacement operations (i.e., inserting a new link) ranged between 3 and 14. The global optimization step can take from 20 minutes to a few hours and is thus clearly beyond interactivity. This step is, however, not disruptive to the design process as it does not require re-iteration.

6.3 Limitations & Future Work

We have presented a method for interactive design of linkage-based characters. Our method takes a purely kinematic approach to simulating the motion of a mechanical assembly. Consequently, there is no notion of friction and mechanical stress on the assembly, and we do not explicitly measure the amount of torque and force required to drive the motion of the character. Nevertheless, we noticed that our criteria for avoiding degeneracies naturally increases the length of the moment arms. In the future, we would like to exchange our kinematic solver with a physics-based simulation approach and explicitly take mechanical efficiency into account during optimization. We also plan to extend our method to design spatial linkage

structures. Since none of the steps in our design system are conceptually restricted to planar mechanisms, we believe this to be a promising direction.

It should be noted that our system does not check for intersections between the different mechanical components during the design process. Rather, as a post-processing step, the components are offset along the direction normal to the motion plane in order to prevent collisions. This layering operation is combinatorial in nature, and there are no guarantees that desirable depth assignments can be found for arbitrary linkage structures. Although we have not encountered any problematic examples in practice, we would like to extend our system to resolve self-intersections during optimization and thus eliminate the need for the post-processing step.

References

- BÄCHER, M., BICKEL, B., JAMES, D. L., AND PFISTER, H. 2012. Fabricating articulated characters from skinned meshes. In *Proc. of ACM SIGGRAPH '12*.
- BEDINI, S. A. 1964. The role of automata in the history of technology. *Technology and Culture* 5, 1, 24–42.
- BICKEL, B., BÄCHER, M., OTADUY, M. A., LEE, H. R., PFISTER, H., GROSS, M., AND MATUSIK, W. 2010. Design and fabrication of materials with desired deformation behavior. *Proc. of ACM SIGGRAPH '10*.
- BICKEL, B., KAUFMANN, P., SKOURAS, M., THOMASZEWSKI, B., BRADLEY, D., BEELER, T., JACKSON, P., MARSCHNER, S., MATUSIK, W., AND GROSS, M. 2012. Physical face cloning. In *Proc. of ACM SIGGRAPH '12*.
- BURMESTER, L. 1888. *Lehrbuch der Kinematik*. Arthur Felix, Leipzig.
- CALÌ, J., CALIAN, D., AMATI, C., KLEINBERGER, R., STEED, A., KAUTZ, J., AND WEYRICH, T. 2012. 3D-printing of non-assembly, articulated models. In *Proc. of ACM SIGGRAPH Asia '12*.
- CEYLAN, D., LI, W., MITRA, N. J., AGRAWALA, M., AND PAULY, M. 2013. Designing and fabricating mechanical automata from mocap sequences. *ACM Transactions on Graphics* 32, 6.
- CHASE, A., 2014. Chase studio, <http://www.andrewchase.com/>. Accessed on April 6, 2014.
- COROS, S., THOMASZEWSKI, B., NORIS, G., SUEDA, S., FORBERG, M., SUMNER, R. W., MATUSIK, W., AND BICKEL, B. 2013. Computational design of mechanical characters. *ACM Trans. Graph.* 32, 4 (July), 83:1–83:12.
- DONG, Y., WANG, J., PELLACINI, F., TONG, X., AND GUO, B. 2010. Fabricating spatially-varying subsurface scattering. In *Proc. of ACM SIGGRAPH '10*.
- FREUDENSTEIN, F. 1954. *Design of Four-link Mechanisms*. Ph. D. Thesis, Columbia University, USA.
- GERGAUT, W., 2011. Mechanische figuren, <https://www.youtube.com/watch?v=rgcaxj55ngw>. Accessed on April 6, 2014.
- HASAN, M., FUCHS, M., MATUSIK, W., PFISTER, H., AND RUSINKIEWICZ, S. 2010. Physical reproduction of materials with specified subsurface scattering. In *Proc. of ACM SIGGRAPH '10*.
- IGARASHI, Y., IGARASHI, T., AND MITANI, J. 2012. Beady: Interactive beadwork design and construction. *ACM Trans. Graph.* 31, 4 (July), 49:1–49:9.
- KAUFMAN, R. E., AND MAURER, W. G. 1971. Interactive linkage synthesis on a small computer. In *Proceedings of the 1971 26th Annual Conference*, ACM, New York, NY, USA, ACM '71, 376–387.
- KING, H. C. 1998. Planar Linkages and Algebraic Sets. Tech. Rep. math.AG/9807023, Jul.
- LAU, M., OHGAWARA, A., MITANI, J., AND IGARASHI, T. 2011. Converting 3D furniture models to fabricatable parts and connectors. In *Proc. of ACM SIGGRAPH '11*.
- MORI, Y., AND IGARASHI, T. 2007. Plushie: An interactive design system for plush toys. In *Proc. of ACM SIGGRAPH '07*.
- PRÉVOST, R., WHITING, E., LEFEBVRE, S., AND SORKINE-HORNUNG, O. 2013. Make It Stand: Balancing shapes for 3D fabrication. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)* 32, 4, 81:1–81:10.
- SKOURAS, M., THOMASZEWSKI, B., BICKEL, B., AND GROSS, M. 2012. Computational design of rubber balloons. In *Proc. of Eurographics '12*.
- SKOURAS, M., THOMASZEWSKI, B., COROS, S., BICKEL, B., AND GROSS, M. 2013. Computational design of actuated deformable characters. *ACM Trans. Graph.* 32, 4 (July), 82:1–82:10.
- STAVA, O., VANEK, J., BENES, B., CARR, N., AND MĚCH, R. 2012. Stress relief: improving structural strength of 3d printable objects. In *Proc. of ACM SIGGRAPH '12*.
- UMETANI, N., IGARASHI, T., AND MITRA, N. J. 2012. Guided exploration of physically valid shapes for furniture design. In *Proc. of ACM SIGGRAPH '12*.
- WANG, W., WANG, T. Y., YANG, Z., LIU, L., TONG, X., TONG, W., DENG, J., CHEN, F., AND LIU, X. 2013. Cost-effective printing of 3d objects with skin-frame structures. *ACM Trans. Graph.* 32, 6, 177:1–177:10.
- WEYRICH, T., PEERS, P., MATUSIK, W., AND RUSINKIEWICZ, S. 2009. Fabricating microgeometry for custom surface reflectance. In *Proc. of ACM SIGGRAPH '09*.
- ZHOU, Q., PANETTA, J., AND ZORIN, D. 2013. Worst-case structural analysis. *ACM Trans. Graph.* 32, 4, 137:1–137:12.
- ZHU, L., XU, W., SNYDER, J., LIU, Y., WANG, G., AND GUO, B. 2012. Motion-guided mechanical toy modeling. *ACM Trans. Graph.* 31, 6 (Nov.), 127:1–127:10.