Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2014), pp. 1–8
Vladlen Koltun and Eftychios Sifakis (Editors)

ChaCra: An Interactive Design System for Rapid Character Crafting

Vittorio Megaro[1] Bernhard Thomaszewski[2] Damien Gauge[1] Eitan Grinspun[3] Stelian Coros[2] Markus Gross[1,2]

[1]ETH Zürich [2]Disney Research Zurich [3]Columbia University

Abstract

We propose an interactive design system for rapid crafting of planar mechanical characters. Our method combines the simplicity of sketch-based modeling with the ease of defining motion through extreme poses. In order to translate these digital designs into fabrication-ready descriptions, our method automatically computes the mechanical structure that makes the characters move as desired. We achieve real-time performance by limiting the mechanical structure between pairs of components to simple building blocks that define, trim, and propagate their motion. By focusing on shape and motion, our system emphasizes the creative aspects of character design while hiding away the intricacies of the underlying mechanical structure. We demonstrate the flexibility of our approach on a set of virtual designs and physical prototypes.

1. Introduction

The art of animation was borne of the desire to breathe life into illustrated characters. With the digital revolution, this artform migrated and transformed via the development of myriad tools, techniques, and codes into the thriving field of computer animation. Today, the field is set against the backdrop of a proliferation of rapid manufacturing devices such as 3D printers and laser cutters, and signs point to yet another revolution, the cyberphysical connecting the digital and physical worlds.

Inexpensive fabrication devices are making it fun to fabricate tangible, physical objects on what could soon become ubiquitous *home 3D printers*. But without motion, these objects are static and lifeless; printed 3D characters serve simply as snapshots, hinting of more to be told. This drive to enhance the expressiveness and storytelling possibilities of printed 3D characters motivates the onset of yet another migration and transformation in our field, as we search for ways to bring to life printed characters. To imbue printed characters with motion, we need visual design tools that assist casual users in expressing their creative visions, hiding or alleviating the myriad technical difficulties that arise in the creation of tangible representations.

To address these goals, our approach draws inspiration from *shadow puppetry*, theatrical plays in which figures are moved behind a screen and lit such as to cast detailed shadows. A beautiful account of this art can be found in the book by Currell [Cur07]. Shadow puppets are planar, rigidly-articulated characters that are often made of plywood or thick cardboard. The pose of a shadow puppet is controlled through a number of rods, which are moved by a puppeteer, invisible to the audience. While this type of *motion control* affords a large space of poses, moving multiple rods in unison to create fluent motion requires skill, experience—and many hands!

Figure 1: *A traditional javanese shadow puppet, actuated with three rods. [sha]*

Inspired by the potential of animated planar shapes and their shadows to feed the imagination, we build on the spirit of shadow puppetry, while trading freedom in posing for simplicity in motion control: instead of using multiple rods, we restrict ourselves to using only a single actuator and build the animation directly into the mechanical structure of the figure. For this purpose, we investigate ways to mechanically connect individual components in an automated way. In order to break down the complexity of this task, we limit the mechanical structure to a small set of parameterized connections, determining the relative motion between components,

as well as mechanisms of fixed structure that delimit and propagate motion throughout the character.

The system that we propose here integrates modeling and animation in a direct and intuitive way that allows casual users to quickly create planar animated characters. Similar to conventional paper crafting, users can get straight to the point by sketching body parts and posing them as desired. Our method then automatically computes the mechanical structure required to make the character move as desired.

By focusing on shape and motion, our system emphasizes the creative aspects of character design while hiding away the intricacies of the underlying mechanical structure. This design metaphor requires a highly responsive interface, which we achieve by building the structure from three simple mechanisms: *connectors*, defining the relative motion between two components, *trimmers*, limiting the motion to a desired range, and *propagators*, transmitting the actuation from its source toward the extremities. Each of these building blocks consists of only two interconnected bars and is thus trivially parameterized. Furthermore, they are designed with the goal of functional orthogonality, which is key for efficient parameter optimization. By breaking down the mechanical structure into these three building blocks, we reduce the difficult problem of general mechanism design to the manageable task of finding parameters for a small number of two-bar structures.

2. Related Work

Physical Character Design A number of works have recently begun to invest the problem of how to translate digital characters into tangible representations. As one of the first works in this direction, Bächer et al. [BBJP12] describe a method to create printable characters from skinned meshes that can be posed in various ways. A similar goal is pursued in the work of Cali et al. [CCA*12]. The method by Prevost et al. [PWLSH13] computes internal mass distributions for printable characters that can balance in surprising poses. Skouras et al. [STC*13] describe a method for designing deformable characters by optimizing for actuation parameters and heterogeneous material distributions such as to obtain a desired range of motion.

Closest to our method are three recent works on the design of mechanical characters. Zhu et al. [ZXS*12] describe a design system for mechanical toys made of articulated parts that can perform rotational and translational motions, as well as simple combinations thereof. Each of the parts is driven by its own mechanism, which are located in a box underneath the character. By contrast, our method integrates the mechanical structure tightly with the character.

The method of Coros et al. [CTN*13] targets mechanical characters with more complex motions. Starting from an articulated character, the user sketches motion curves to indicate how different parts should move. The method then automatically selects a mechanism from a library of predesigned templates and computes its parameters in order to best approximate the desired motion. As a chief difference, the method of Coros et al. requires an articulated 3D character with the right number of degrees of freedom as input, whereas in our system, the user directly designs the character as well as its motion. Both approaches have their justification. If a digital character is available, it is convenient to use this representation directly. Otherwise, however, modeling a 3D character that satisfies the input constraints is a difficult task beyond the capabilities of an average user. With our system, even casual users can design animated characters in a matter of minutes.

Using motion capture data as input, Ceylan et al. [CLM*13] automatically create mechanical representations of humanoid characters that closely approximate the target motion. Similar to our method, they use specific compound mechanisms, so called oscillation modules, in order to drive the relative motion between rigid links and to propagate rotational motion throughout the character. However, based on gears, pulleys, and four-bar linkages, these oscillation modules are very different from the mechanisms used in our work. Furthermore, instead of relying on motion capture data to determine the character's movements, our method takes an intuitive approach to motion design based on extreme poses, which are easy to create and edit.

Design Systems for Arts and Crafts Our work is inspired by existing methods that assist the designer in creating craft- and artwork such as paper models [MI07], popup architecture [LSH*10], or beadwork [IIM12]. As one particular example, the method by Mori and Igarashi [MI07] allows its users to interactively create plush toys by sketching their deformed geometry while the flat panels required for fabrication are computed automatically. Our system takes a similar sketch-based approach to geometry and topology design, but the computational problem of mechanism design is very different from plush toy design.

Fabrication-Oriented Geometry Processing Due to the increasing availability of rapid manufacturing devices, there has recently been an increasing interest in the various challenges related to fabricating 3D geometry. For instance, several works [SVB*12, ZPZ13, WWY*13] analyze and improve the structural stability of static models. As another example, the work by Luo et al. [LBRM12] offers an automatic way of decomposing large objects into printable chunks. While our method is not limited to a particular manufacturing process or technology, the characters that we design consist only of planar parts which can be manufactured in a fast and cost-efficient way using a laser cutter.

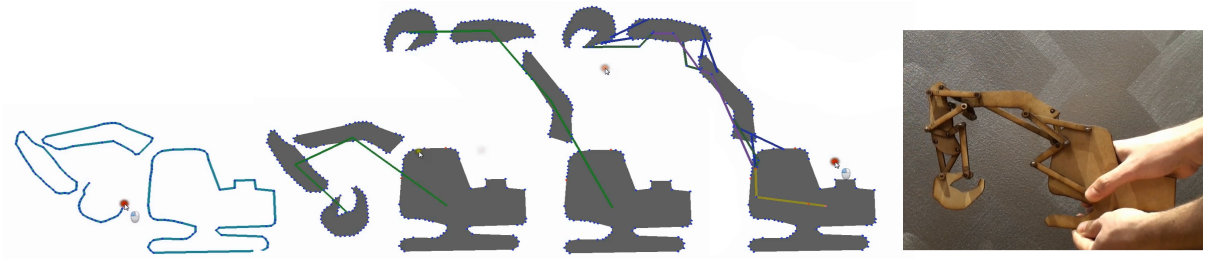

Figure 2: *An overview of our design system: the user sketches geometry (1) and defines the range of motion (2 and 3). Our method generates the mechanical structure (4) that is used to build a physical prototype (5).*

3. Method Overview

Our system enables casual users to quickly create mechanical characters that perform periodic motions. The different steps of our design process are summarized in Fig. 2. The user starts by sketching the individual parts of the character and indicating which components to connect. Connections are specified by roughly delineating the regions of adjacent parts that should be attached. With its geometry and topology defined, the desired range of motion of the character is specified through two poses per extremity, corresponding to turning points in the animation. Our method then automatically computes the mechanical structure that connects the individual components in a way that, when driven by a central actuator, the character moves between the extreme poses as desired. It is worth noting that the entire design process is very fast, allowing for quick exploration of various design choices.

Before we explain our method in more detail, we will briefly describe the representation that we use to model and animate our characters.

3.1. Representation and Computational Model

A character consists of a number of rigid components C_i, each of which is defined by three degrees of freedom $\mathbf{s}_i \in \mathbb{R}^3$, orientation $\theta \in \mathbb{R}$ and position of center of mass $\mathbf{t} \in \mathbb{R}^2$. We will use $R(\theta)$ to denote the 2×2 rotation matrix corresponding to θ. Furthermore, we define the transformation $\mathbf{x}$ from a point $\mathbf{p}^l$ in local coordinates to its corresponding point $\mathbf{p}^w$ in world-space as

$$\mathbf{p}^w = \mathbf{x}(\mathbf{s}_i, \mathbf{p}^l) = R(\theta_i)\mathbf{p}^l + \mathbf{t}_i \; . \tag{1}$$

For simplicity, we will omitt the superscripts if they are clear from the context. The rigid components are linked through three types of connections, each of which consists of two bars. We model these bars using distance constraints of the form $C_{ij}(\mathbf{p}_i, \mathbf{s}_i, \mathbf{p}_j, \mathbf{s}_j) = ||\mathbf{x}(\mathbf{p}_j, \mathbf{s}_j) - \mathbf{x}(\mathbf{p}_i, \mathbf{s}_i)|| = l_{ij}$ and assemble all of the constraints into a global vector $\mathbf{C}$. Additionaly, the component corresponding to the main body is fixed and there is a gear attached to it that is driven by a single motor. When stepping the orientation of this gear over time, the configuration $\mathbf{s}$ of the assembly is obtained by minimizing the constraint energy $E(\mathbf{s}) = \mathbf{C}(\mathbf{s})^t\mathbf{C}(\mathbf{s})$ using Newton's method. By construction, the constraints fully define the state of all components and there are no conflicts among them such that the constraint energy vanishes at its minimum.

4. Automatic Structure Design

Given the geometry for all body parts C_i and two extreme poses $\mathbf{s}^a$ and $\mathbf{s}^b$ as input, our method automatically computes the mechanical structure that makes the character move as desired. There are three types of building blocks in this structure. Between each pair of connected components C_i and C_j there is a *connector*, defining how the components move relative to each other, a *trimmer* that limits the relative motion between the components to a desired range, and a *propagator*, transmitting motion from one component to the next. We will describe each of these building blocks in the following.

4.1. Connections

In order to provide enough room for artistic freedom when designing the range of motion of a character, our method supports three different types of connections between rigid body parts. Fig. 3 illustrates these three connection types and shows examples of the different motions that they induce.

A connector consists of two bars, each of whose endpoints is attached to one of the components. The relative positions of these endpoints define the type of the connection. We require that each connector leaves exactly one degree of freedom for relative motion between the two components. A simple counting argument shows that this requirement is satisfied for the parallel and the cross coupling connectors. This argument remains valid as the attachment points on one component converge to a single point, yielding the pin coupling. However, with the attachment points collapsing on both sides, the resulting structure exhibits two degrees of freedom and is thus not a valid connector.

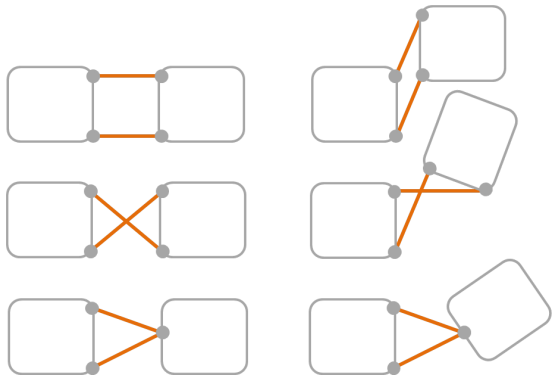

Figure 3: *Different ways of connecting two building blocks (from* top *to* bottom*): parallel, cross, and pin couplings in their default (*left*) and deformed configurations (*right*).*

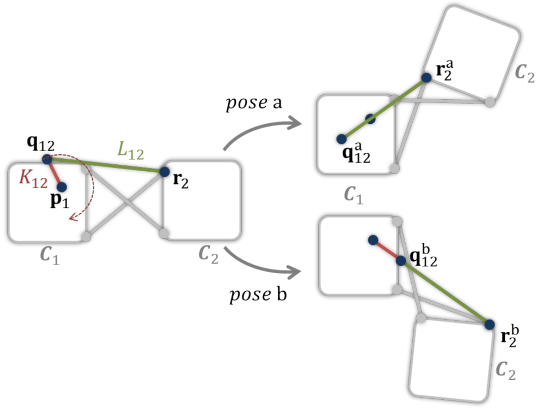

Figure 4: *Trimming mechanism consisting of two bars K_{12} and L_{12}. When rotating K_{12} around $\mathbf{p}_1$ as indicated, C_2 oscillates between the two extreme poses as shown on the right.*

Automatic Computation of Connection Parameters Instead of asking the user to manually specify the connection parameters for a given pair of components, we infer the type of connection as well as the corresponding attachment points automatically from the two extreme poses provided by the user. As shown in Fig. 2, the user indicates through sketching which components to connect and which regions on these components should be considered for attaching the connector. While the user creates the first pose, we impose no restriction as to the relative position and orientation of the two components. However, while the user translates and orients the component to create the second pose, our method works in the background to determine whether the current pose can be reasonably well approximated and, if so, which type of connection is best suited. We take a sampling-based approach in order to answer these questions in a timely manner. For the regions on the two components indicated by the user, we create sets of regularly-spaced sample points. We then randomly select pairs from these two sets and compute their distances d^a and d^b in the two poses $\mathbf{s}^a$ and $\mathbf{s}^b$, respectively. We generate a fixed number (we use 50) of these pairs and order them according to the smallest distance variation d^a/d^b. We then process the list in order to find two edges with as low as possible a variation in distance while ruling out degenerate cases such as a parallel couplings that fold over to a cross coupling. This entire process is very fast such that we can provide interactive visual feedback on the feasibility of the second pose while the user drags on the component.

4.2. Motion Trimming

Once two body parts C_1 and C_2 are attached through a connector, there is only one degree of freedom left between them. Without restricting generality, we will assume that the first component, C_1, is fixed such that the motion of the system is confined to a one-dimensional curve in the state space of the second component C_2. We note that this curve is closed, i.e., moving the second component along its single degree of freedom through state space will eventually lead to it coming back to its initial configuration. In practice, however, we want to *trim* this motion to the part between the user-provided extreme poses. To this end, we insert a *trimmer* between the two components, consisting of two interconnected bars, K_{12} and L_{12}. As illustrated in Fig. 4, one end point of K_{12} connects to a point on C_1 with local coordinate $\mathbf{p}_1$, one endpoint of L_{12} connects to C_2 in $\mathbf{r}_2$, and the two remaining endpoints of the bars are connected to each other. By rotating K_{12} around $\mathbf{p}_1$, i.e., prescribing its orientation relative to C_1, we can drive the motion of C_2.

Let $\theta \in [\theta^a, \theta^b]$ denote a scalar that parameterizes the motion of the character between its two extreme poses $\mathbf{s}^a$ and $\mathbf{s}^b$. Furthermore, suppose that we can find points $\mathbf{p}_1$ and $\mathbf{r}_2$ such that their world-space distance

$$d(\theta) = ||\mathbf{x}(\mathbf{s}_2(\theta), \mathbf{r}_2) - \mathbf{x}(\mathbf{s}_1(\theta), \mathbf{p}_1)||_2 \qquad (2)$$

increases monotonically between the two poses, i.e., $d(\theta^a) < d(\theta) < d(\theta^b)$. Then we can determine the lengths $l(K_{12})$ and $l(L_{12})$ of the two bars such that

$$\begin{aligned} d(\theta^a) &= l(L_{12}) - l(K_{12}) \,, \\ d(\theta^b) &= l(L_{12}) + l(K_{12}) \,. \end{aligned}$$

By inserting these two bars, we mechanically impose bound constraints on the world-space distance between $\mathbf{p}_1$ and $\mathbf{r}_2$. When driving the orientation of K_{12} such that it performs a full revolution, this distance will vary between the two extremal values d^a and d^b, hitting each of them exactly once. This trimming structure will induce a motion in C_2 satisfying three properties: first, because the extremal distances d^a and d^b are reached, C_2 will attain the extreme poses since, due to the monotonicity of $d(\theta)$ for $\theta \in [\theta^a, \theta^b]$, there is a one-to-one map between the distance $d(\theta)$ and the state

$\mathbf{s}_2(\theta)$ of the second component. Second, the motion will switch directions at the extreme poses since going beyond them would require $d(\theta)$ to increase above or fall below the maximum or minimum distance, respectively. Finally, the motion between the two poses is monotonic, since $d(\theta)$ is monotonic.

Monotonicity In the above construction, we assumed that the distance between $\mathbf{p}_1$ and $\mathbf{r}_2$ varies monotonically between the extreme poses. In order to find two points that satisfy this requirement, we take a sampling approach similar to the one used for determining the parameters of the connections. From a discrete set of sample points on the two components, we want to find pairs that yield a comparatively large variation in the corresponding distance values for the extreme poses, d^a and d^b. We find candidate pairs through random sampling with a probability biased towards selecting points close to the center of mass of C_1 and close to the border of C_2 facing C_1. We collect a fixed number of such pairs (we use 50) and order them according to the resulting distance variation d^a/d^b. We then check whether the first pair leads to a monotonic increase in distance. We perform this test by computing distance values for a small number (we use 20) of configurations distributed evenly between the extreme poses and check whether the resulting sequence is monotonic. If this is the case, we keep the pair and return, otherwise we continue with the next pair in the list.

4.3. Motion Propagation

With a trimmer and connector placed between a given pair of components C_1 and C_2, their relative motion is fully defined. We could now attach a motor to the trimmer bar K_{12} in order to drive the motion of C_2 as indicated in Fig. 4. However, in order to propagate motion further downstream, i.e., drive an additional component C_3 through C_2, we need to relay the actuation signal from the motor in order to create a source of rotational motion on C_2. This is the purpose of the propagation mechanism.

Propagation Mechanism We apply the same principle that was used for the trimming structure, but with inverted roles for the two components. As illustrated in Fig. 5 (*top*) this propagation structure consists again of two bars, M_{12} and N_{12}, whose endpoints are connected at points $\mathbf{p}_2$ and $\mathbf{r}_1$ to C_2 and C_1, respectively. Again, by finding $\mathbf{p}_2$ and $\mathbf{r}_1$ such that the distance between the world-space points is monotonic between the two extreme poses, we can compute the lengths of the two bars from collinearity conditions. Note that the collinearities have to be attained at the exact same configuration as for the trimming structure since otherwise, the motion would be affected or the structure might lock. With this propagation mechanism in place, the bar M_{12} will be passively actuated by K_{12}, i.e., rotating K_{12} around $\mathbf{p}_1$ will induce a corresponding rotation of M_{12} around $\mathbf{p}_2$.

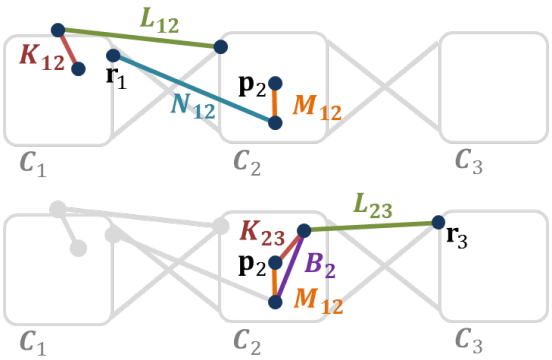

Figure 5: Top*: a propagator (M_{12}, N_{12}) is added to transmit the actuation signal from C_1 to C_2.* Bottom: *in order to drive C_3, the actuation signal is fed into another trimmer (K_{23}, L_{23}) that is rigidly connected to the propagator through B_2.*

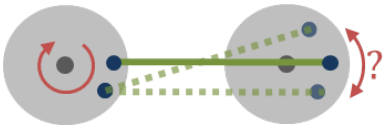

Figure 6: *A singular configuration in the driven wheel* (right) *arises as its center becomes collinear with the two end points of the bar. When rotating the driving wheel* (left) *further, the right end point of the bar could move either up or down as indicated.*

Connecting Propagators to Trimmers With the trimming and propagation mechanisms defined, it is now straightforward to add a third component C_3 (and more) to the system. As illustrated in Fig. 5 (*bottom*), we pick up the actuation signal from M_{12}, which is passively actuated by the propagation structure between C_1 and C_2. We insert two bars, K_{23} and L_{23}, in order to create the trimming structure between C_2 and C_3. As the first attachment point for K_{23}, we choose $\mathbf{p}_2$ and it remains to find $\mathbf{r}_3$ on C_3 such that, again, a monotonicity relation in the world-space distance between $\mathbf{p}_2$ and $\mathbf{r}_3$ is observed. The lengths of the two bars are then determined using the collinearity conditions (3). Finally, in order to transmit the rotation of M_{12} to K_{23}, we rigidly link the two bars with (yet) another bar B_2.

Limitations Both the motion trimmer and propagation mechanisms described above rely on the principle of extremizing the distance between pairs of points on the connected components. These extreme distances are assumed whenever the two bars of the corresponding mechanism align or, equivalently, its three points become collinear. While this collinearity poses no problem for an actively driven trimming structure, it creates a singular configuration for the propagation structure, which is driven passively by definition.

This situation is analogous to the example shown in Fig. 6, which illustrates that singularities inevitably arise when using a kinematic constraint to transmit rotational motion from a driving wheel to a driven wheel. In the absence of regularizing effects (inertia, gravity), the rotation direction of the driven wheel is not unique and, as a result, the wheel could rotate either way or the system might lock altogether.

We will revise the trimming and propagation mechanism in order to avoid singularities. However, the key insights described in this section, i.e., how to limit relative motion between connected components and how to propagate actuation, are equally applicable in this revised setting.

5. Avoiding Singularities

In order to avoid singularities *ab initio*, we abandon the actuation in terms of full revolutions and instead directly generate oscillatory motions for the bars K_{jk} and M_{ij} on each component C_j.

Propagation Mechanism We consider again the propagation mechanism shown in Fig. 5, but instead of asking for the bars M_{12} and N_{12} to become collinear, we require them to stay at a safety angle of α away from collinearities. We will use a geometric construction in order to compute lengths for the two bars that fulfill this requirement.

We assume that the first component remains fixed and that the user provided extreme poses for C_2. We thus know the attachment point $\mathbf{r}_1$ for N_{12}, which is the same in both poses, as well as the world-space positions $\mathbf{p}_2^a$ and $\mathbf{p}_2^b$ of the attachment point for M_{12} in the two poses, respectively. We start by defining two vectors $\mathbf{l}^a = \mathbf{p}_2^a - \mathbf{r}_1$ and $\mathbf{l}^b = \mathbf{p}_2^b - \mathbf{r}_1$, indicated as dashed lines in Fig. 7, and require that M_{12} form angles of α and $\pi - \alpha$ with $\mathbf{l}^a$ and $\mathbf{l}^b$ in the two poses. In order to compute the lengths of M_{12} and N_{12} satisfying these conditions, we parameterize the world-space positions of the connection between the bars using a scalar s such that

$$\mathbf{q}_{21}^a(s) = \mathbf{p}_2^a + s\mathbf{e}^a \,,$$
$$\mathbf{q}_{21}^b(s) = \mathbf{p}_2^b + s\mathbf{e}^b \,,$$

where $\mathbf{e}^a$ and $\mathbf{e}^b$ are unit vectors forming angles of α and $\pi - \alpha$ with the lines $\mathbf{l}^a$ and $\mathbf{l}^b$, respectively. For any values of s, the length of M_{12} is the same in both configurations by construction, but the length of N_{12} can differ. We can thus determine s by requiring the lengths of N_{12} to be equal in both configurations, which—using simple trigonometric relations—yields

$$s = \frac{(d^a)^2 + (d^b)^2}{2\cos(\alpha)(d^a + d^b)} \,. \qquad (3)$$

Actuation Mechanism With the propagation mechanism now safe from singularities, we revise the actuation mechanism correspondingly. As illustrated in Fig. 9, M_{12} does not perform full revolutions around $\mathbf{p}_2$, but only describes a certain angle β. In order to drive the motion of C_3 through C_2, we first find a point $\mathbf{p}_3$ on C_3 such that the distance between $\mathbf{p}_2$ and $\mathbf{p}_3$ varies monotonically between the two extreme poses. We then connect a bar K_{23} to $\mathbf{p}_2$ and attach it rigidly to M_{12} at an angular offset γ. From β and γ we know the orientation of K_{23} in both configurations. We can thus compute the length of K_{23} in direct analogy to (3). The choice of the offset angle γ is governed by the requirement to avoid collinearity between K_{23} and L_{23}. As illustrated in Fig. 8, we can derive upper and lower bounds on γ such that no singularity is crossed. In order to stay as far away as possible from either of them, we choose the average angle $\gamma = \frac{1}{2}(\gamma_{min} + \gamma_{max})$.

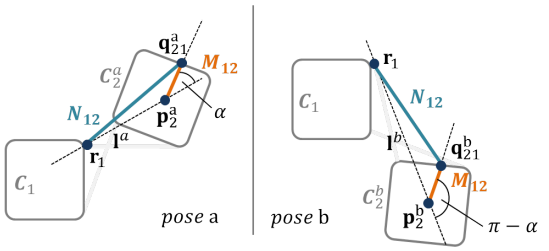

Figure 7: *To safeguard against collinearities, we require N_{12} and M_{12} to maintain an angle of α in the two extreme poses.*

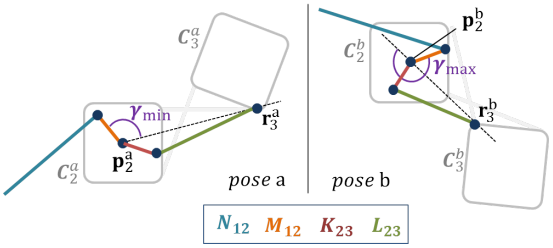

Figure 8: *In order to avoid collinearities in the trimming structure, the angular offset γ between M_{12} and K_{23} has to be in the range $(\gamma_{\min}, \gamma_{\max})$.*

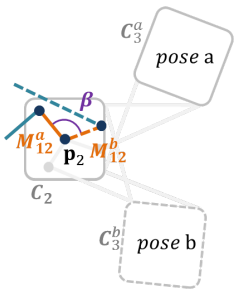

Figure 9: *Definition of β*

Optimizing Moment Arms Up to now, we have considered α as a means of safeguarding against singularities. Intuitively, α should provide sufficient margins to warrant robustness in the presence of inaccuracies due to fabrication. Crucially, there is another important aspect that influences the choice of α, an aspect that we discovered by virtue of actual fabrication:

As depicted in Fig. 7, the distance from $\mathbf{q}_{12}^a$ to the line $\mathbf{l}^a$ corresponds to a *moment arm*, whose length determines how much force needs to be applied through bar N_{12} in order to obtain a given torque in $\mathbf{p}_2^a$. It is worth noting that, by construction, the moment arms are equal in both configurations and we refer to this value as $\tilde{m} = \sin(\alpha)s$. Let $m(\theta)$ be a parameterization of the moment arm between the two extreme poses. It is clear from Fig. 7 that $m(\theta)$ is bounded from below as

$$m(\theta) > \tilde{m} \quad \text{for} \quad \theta \in (\theta^a, \theta^b). \tag{4}$$

Using (3) it can be seen that $\tilde{m}$ is proportional to $\tan(\alpha)$ such that we can increase the moment arm for the entire range of motion by increasing α. In practice, larger moment arms are desirable as they will generally reduce the amount of force required to drive the motion of the character. On the other hand, a larger α means less motion for the propagator, which will effectively require a longer K_{23} in order to create the same range of motion for C_3. In our experiments, we obtained good results for values of α between 20 and 50 degrees.

6. Results and Discussion

We demonstrate the versatility of our method by designing a range of animated mechanical characters, two of which we fabricate. The motion of the characters, as well as screen captures of the design process, are presented in the accompanying video. Designing each one of the characters we show took less than a couple of minutes. In addition, we found that building our method around the familiar sketching and posing metaphors resulted in a very intuitive and easy-to-use computational design system.

As we illustrate with several of the characters we design, our method is capable of transmitting the motion of the input driver to long kinematic chains. The neck assembly in the Dragon figure (Fig. 10), for instance, is composed of an initial set of 6 rigid bodies. The final mechanism comprises a total of 30 components. In principle there is no limit on the length of the kinematic chains that can be animated with our method. However, we noticed that errors due to mechanical play and material deformations can limit the complexity of the structures that we can create in practice.

Our method is developed for the purpose of controlling the motions of serial kinematic chains. However, digital characters are typically best represented using tree, or branching structures. The Ballerina (Fig. 11) example, which is representative of the typical complexity expected for planar characters, shows that full-body motions can still be synthesized with our method by treating each limb independently of the others. The input drivers for each limb are placed on the main bodies of the characters, and, if needed, they can be connected to one another via gear trains [CTN*13].

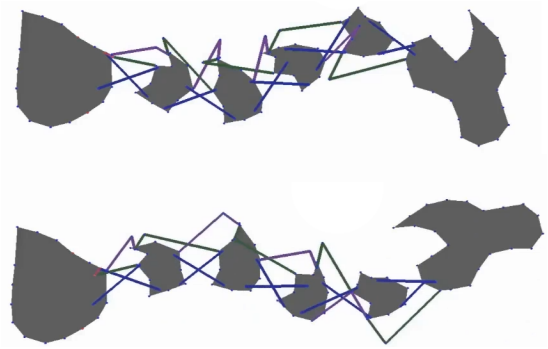

Figure 10: *Two views of the* Dragon Neck *example.*

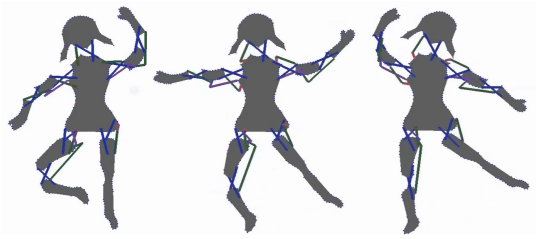

Figure 11: *Three views of the* Ballerina *character.*

Fabrication The mechanical characters we design lend themselves naturally to fabrication using planar components. Laser-cutting is therefore our method of choice for creating physical prototypes, but 3D printing is of course a viable alternative. Once the design of a character is completed, there are a few additional steps that need to be performed before the character is ready for fabrication. In particular, we need to ensure that the movement of the individual mechanical components is collision free. To this end we take a layering approach, where we offset each component by a different amount along the normal of the motion plane. In a general setting, automatically determining the offset for each component is very complex (see Coros et al. [CTN*13]). However, one advantage of our approach to designing animated mechanical characters is that we know in advance which mechanical components can potentially interfere with each other. We can thus solve the layering problem by offsetting by a fixed amount each of the six bars between every pair of connected components.

The kinematic properties of the finalized mechanical designs are used to procedurally generate 2-dimensional curves that prescribe the geometry of the components, including the circular slots we need for articulations. We use metal pins for the joints and 4mm plywood plates for the body parts, which are fabricated using a laser cutter. Going from a digital design to a fabricated prototype took about 2 hours for the Excavator model (Fig. 2). Roughly half the time was needed to laser cut all the components, and half the time to assemble

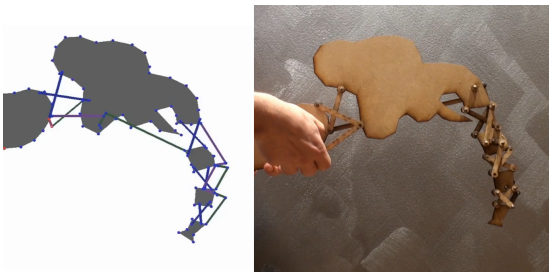

Figure 12: *The digital* Elephant *character (left) and the fabricated prototype (right).*

the prototypes. As demonstrated by this example, the motions of the manufactured prototypes generally match well the motions of their digital counterparts. However, we do not explicitly take into account the forces needed to drive the motion of our characters. Consequently, deformations of the lead-bearing mechanical components, amplified by the mechanical play at the joints can, in some designs, lead to characters that exhibit a reduced range of motion, as our Elephant character (Fig. 12) illustrates.

7. Discussion, Limitations and Future Work

We proposed an interactive design system that allows casual users to employ sketching and posing operations in order to create animated mechanical characters. The technical aspects of our system were significantly influenced by aspects pertaining to fabrication of tangible, physical characters. Indeed, we found that closely integrating fabrication into our research process was vital to developing a practical, effective system. For example, the need for optimization of moment arms became blatantly apparent after our initial physical prototypes failed: while virtual characters can withstand arbitrary torques and forces, physical artefacts typically have non-negligible limitations.

Although our system is very intuitive to use, this comes at the cost of not having fine-level control over all aspects of the resulting motions: specifying only the extreme poses does not imply that the intermediate motions can be exactly prescribed. This provides an interesting avenue for future work. We envision a design system where the user can refine the motions of the characters by providing intermediate target poses that guide the animation as needed.

Our design methodology currently takes only kinematic considerations into account. This allows our system to be highly interactive, providing immediate feedback to the users, and allowing them to easily explore different designs. However, when building the physical prototypes of the digital characters, the force needed to drive their motion can be too high, leading to material deformations and wear-and-tear in the mechanical components. In the future, we plan to also analyze the stresses induced in the mechanical components, and refine the designs so that they are minimized.

References

[BBJP12] BÄCHER M., BICKEL B., JAMES D. L., PFISTER H.: Fabricating articulated characters from skinned meshes. In *Proc. of ACM SIGGRAPH '12* (2012). 2

[CCA*12] CALÌ J., CALIAN D., AMATI C., KLEINBERGER R., STEED A., KAUTZ J., WEYRICH T.: 3D-printing of non-assembly, articulated models. In *Proc. of ACM SIGGRAPH Asia '12* (2012). 2

[CLM*13] CEYLAN D., LI W., MITRA N. J., AGRAWALA M., PAULY M.: Designing and fabricating mechanical automata from mocap sequences. *ACM Transactions on Graphics 32*, 6 (2013). 2

[CTN*13] COROS S., THOMASZEWSKI B., NORIS G., SUEDA S., FORBERG M., SUMNER R. W., MATUSIK W., BICKEL B.: Computational design of mechanical characters. *ACM Trans. Graph. 32*, 4 (2013). 2, 7

[Cur07] CURRELL D.: *Shadow Puppets & Shadow Play*. The Crowood Press, 2007. 1

[IIM12] IGARASHI Y., IGARASHI T., MITANI J.: Beady: Interactive beadwork design and construction. *ACM Trans. Graph. 31*, 4 (2012). 2

[LBRM12] LUO L., BARAN I., RUSINKIEWICZ S., MATUSIK W.: Chopper: Partitioning models into 3d-printable parts. *ACM Trans. Graph. 31*, 6 (2012). 2

[LSH*10] LI X.-Y., SHEN C.-H., HUANG S.-S., JU T., HU S.-M.: Popup: Automatic paper architectures from 3d models. *ACM Trans. Graph. 29*, 4 (2010). 2

[MI07] MORI Y., IGARASHI T.: Plushie: An interactive design system for plush toys. In *Proc. of ACM SIGGRAPH '07* (2007). 2

[PWLSH13] PRÉVOST R., WHITING E., LEFEBVRE S., SORKINE-HORNUNG O.: Make it stand: Balancing shapes for 3d fabrication. In *Proc. of ACM SIGGRAPH '13* (2013), pp. 81:1–81:10. 2

[sha] http://www.ssplprints.com/image/87440/javanese-shadow-puppet-early-20th-century. 1

[STC*13] SKOURAS M., THOMASZEWSKI B., COROS S., BICKEL B., GROSS M.: Computational design of actuated deformable characters. *ACM Trans. Graph. 32*, 4 (2013). 2

[SVB*12] STAVA O., VANEK J., BENES B., CARR N., MĚCH R.: Stress relief: improving structural strength of 3d printable objects. In *Proc. of ACM SIGGRAPH '12* (2012). 2

[WWY*13] WANG W., WANG T. Y., YANG Z., LIU L., TONG X., TONG W., DENG J., CHEN F., LIU X.: Cost-effective printing of 3d objects with skin-frame structures. *ACM Trans. Graph. 32*, 6 (2013), 177:1–177:10. 2

[ZPZ13] ZHOU Q., PANETTA J., ZORIN D.: Worst-case structural analysis. *ACM Trans. Graph. 32*, 4 (2013), 137:1–137:12. 2

[ZXS*12] ZHU L., XU W., SNYDER J., LIU Y., WANG G., GUO B.: Motion-guided mechanical toy modeling. *ACM Trans. Graph. 31*, 6 (2012). 2