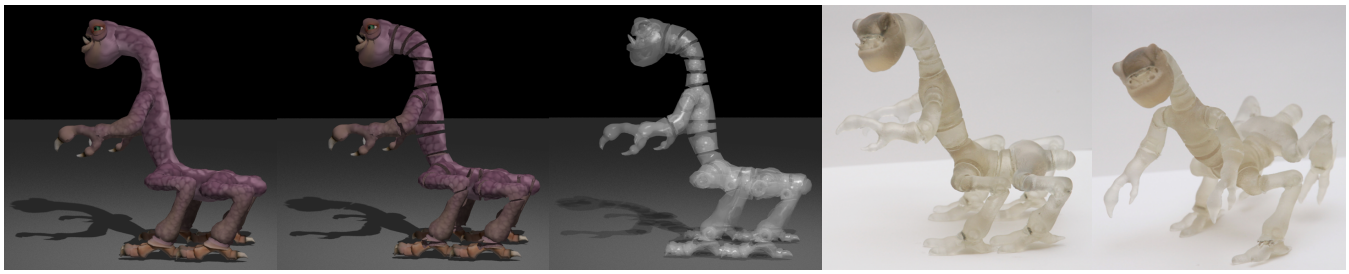
Fabricating Articulated Characters from Skinned Meshes

Moritz Bächer
Harvard University

Bernd Bickel
TU Berlin

Doug L. James
Cornell University

Hanspeter Pfister
Harvard University



(a) (b) (c) (d) (e)

Figure 1: *Given a skinned mesh (a), we estimate (b) a fabricatable articulated character with (c) internal joints of hinge and ball-and-socket type. (d,e) Final 3D printed characters (transparent material) have durable joints with a frictional design for character posing.*

Abstract

Articulated deformable characters are widespread in computer animation. Unfortunately, we lack methods for their automatic fabrication using modern additive manufacturing (AM) technologies. We propose a method that takes a skinned mesh as input, then estimates a fabricatable single-material model that approximates the 3D kinematics of the corresponding virtual articulated character in a piecewise linear manner. We first extract a set of potential joint locations. From this set, together with optional, user-specified range constraints, we then estimate mechanical friction joints that satisfy inter-joint non-penetration and other fabrication constraints. To avoid brittle joint designs, we place joint centers on an approximate medial axis representation of the input geometry, and maximize each joint's minimal cross-sectional area. We provide several demonstrations, manufactured as single, assembled pieces using 3D printers.

Keywords: fabrication, additive manufacturing, animation, articulated solid models.

Links: DL PDF

1 Introduction

Skinned characters are among the most widespread models in computer graphics and have received tremendous attention in recent decades. Skilled artists have years of experience in creating weighted associations between a hierarchical set of bones (*rig*) and groups of vertices on the character's mesh (*skin*). Content creation systems, such as the one built into SPORE [Hecker et al. 2008], allow even naive users to create sophisticated skinned characters.

Recently, online services such as Shapeways have become available, making personalized manufacturing on cutting edge AM technologies accessible to a broad audience. Affordable desktop printers will soon take over, enabling people to fabricate custom-made 3D models at home. However, animation software packages such as Maya or Blender lack a "3D print button" to facilitate converting a virtual articulated model into a fabricatable format. While tools and services that map static properties such as geometry and appearance exist, the articulated behavior—a key property of posable skinned models—remains unmapped.

In this paper, we present a technique that estimates an articulated character model suitable for manufacturing with AM technologies from a given skinned mesh (see Fig. 1 (a)). Our method is capable of generating posable models consisting of a set of piecewise rigid pieces with non-overlapping, physically meaningful ball-and-socket or hinge joint parts (Fig. 1 (b,c)).

Note that a direct mapping from virtual articulated to manufacturable, jointed models does not exist. For starters, rig joints are close to physically meaningless as they can move out of the deformed geometry as illustrated in Fig. 2 left with a rigged cylinder. Furthermore, because they are also not guaranteed to be embedded in the character's geometry in its rest pose, they are not a reliable estimate for joint center placement. Also, while rig joints are zero-dimensional points, mechanical joints are volumetric entities that need to be large enough for structural strength, and as such can potentially "collide" with each other if care is not taken in the joint design process (see Fig. 2 right). Our approach addresses these concerns.

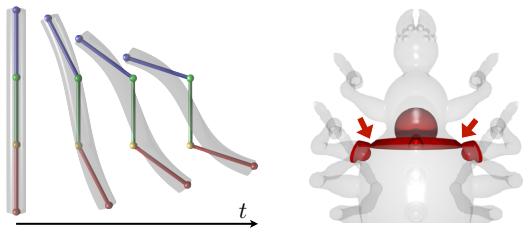


Figure 2: ***Virtual Rig vs. Mechanical Joints:*** *When animating a rigged cylinder (left), we observe that the rig joints do not fall together with actual rotation centers and move out of the deformed geometry. (right) If we maximize the individual sizes of mechanical joints (and thus their strength), they could collide (red).*

While our method is capable of automatically generating articulated models with ball-and-socket joints set to default ranges, these 3-DOF (degrees of freedom) defaults may restrict the posing space of fabricated characters either too little or too much. We therefore allow users to switch individual joints to hinge type (1-DOF) and to specify range parameters differing from defaults for both of our joint designs. For all our demonstrations, user-intervention is limited to a subset of the joints.

After first analyzing the mesh and skinning weights, we estimate proxy joint locations, and assign custom parametric models for volumetric joint geometry that are consistent with any user-specified joint limits. We then proceed to optimize joint parameters (location, size, etc.) to increase joint strength while avoiding overlapping joint geometry. By augmenting our joint models with tiny bumps to increase joint friction, our output models can be posed and will retain their configuration (see Fig. 1 (d,e)). Finally, the estimated joints are carved out of the character mesh using CSG operations. Additional overviews of our approach are given in Fig. 3 and § 3.

For completeness and to assure high quality of our output models, we approximate the characters' surface appearance also. Because the resolution of the geometry of many skinned characters is kept low for fast rendering, we estimate microgeometric detail from normal maps if available. Carving out joints from character meshes also works on textured content. We demonstrate the applicability of our approach on a number of examples (see Fig. 1, 11, 12, 13).

We show that an analysis of skinning weights leads to a plausible segmentation of the character's geometry into rigid body parts. Furthermore, we present novel, geometric approximate models of joint strength, that, together with our method to avoid joint-joint collisions, ensure strong and functional joints in our output models. Also, our collision resolution allows us to keep as much of the "fabricatable" input articulation in our posable output models as possible. To the best of our knowledge, we are the first to present a technique to automatically convert skinned meshes into durable, articulated models.

2 Related Work

We estimate a piecewise-rigid, jointed volume model suitable for 3D fabrication from an input character whose articulation is encoded in its skin. Articulated characters are widespread in computer animation, with linear blend skinning (LBS) and example-based approaches common [Lewis et al. 2000; Mohr and Gleicher 2003; Kavan et al. 2008]. Most character rigging methods either estimate a skeleton or LBS from a mesh [Baran and Popović 2007] or estimate a skinned character model from example poses [Kry et al. 2002; Mohr and Gleicher 2003; Wang et al. 2007] or input animations [James and Twigg 2005]. We focus on articulation specified as a linear blend skin as it is the most widely used format. However, current AM techniques do not support printing of skinned meshes.

Recent advances in AM and fabrication of CG content have addressed the conversion of virtual furniture models to fabricatable parts [Lau et al. 2011], and the approximation of an object's microgeometry [Weyrich et al. 2009], spatially-varying reflectance [Matusik et al. 2009], subsurface scattering [Dong et al. 2010; Hašan et al. 2010], and user-specified deformation behavior [Bickel et al. 2010]. For skinned characters, however, existing tools only convert their appearance and shape properties and ignore their articulation.

Because our targeted output models share strong similarities with articulated toys such as dolls or puppets, and action figures, we draw inspiration from the large body of patents filed on this topic. They describe many mechanical joints ranging from basic swivel to elaborate, multi-part designs [Abbat 1993; Ferre 2000] that overcome common structural and range shortcomings. However, none of them is based on a geometric model of joint strength that complies with range constraints like our hinge and ball-and-socket designs. To make our joints posable, we fabricate small protrusions similar to [Grey 1999; Wai 2006] that cause friction under joint motion but extent their ideas to prevent fusion during manufacturing.

When recasting our joint optimizations as pure geometric problems, we draw inspiration from structural engineering [Beer et al. 2011]: to increase the strength of a simple structure, civil engineers identify and maximize its critical cross-sectional area. In graphics, similar ideas have been used to automate the generation of truss structures [Smith et al. 2002] and procedural models of buildings [Whiting et al. 2009].

3 Overview

For articulated characters, we have to successfully map three components from the virtual model to reality: two static properties, namely *geometry* and *appearance*, and the model's *articulation* that allows it to be posed. See Fig. 3 for an overview of our fabrication pipeline. Next, we identify the properties we use.

3.1 Input: Skinned Characters

The input to our estimation process is a skinned character (see Fig. 3 left). The input *geometry* is specified as semi-organized set of oriented face tuples $\mathbf{f} \in F$ whose components index into a set of vertices $\mathbf{v} \in V$. Optionally, *appearance* is specified with color information provided as diffuse texture, and micro-geometric detail encoded in a normal map. As indicated in Fig. 3 (d), our input mesh could potentially consist of a set of individual, overlapping mesh components. By repairing (removing duplicate vertices, resolving violations of manifoldness, etc.) and unifying this set of components, we compute a manifold, closed surface mesh $(V_{\mathcal{F}}, F_{\mathcal{F}})$. Because this mesh fulfills the requirements of manufacturing, we call it a fabrication mesh $\mathcal{F}$. Without loss of generality, we hereafter assume faces and vertices to refer to entities of repaired meshes, and the faces to be triangles.

The *articulation* behavior is specified by a LBS model wherein each vertex i in V is weighted to link $l \in L$ by a (nonnegative) skinning weight w_{il}, such that the deformed vertex position is given by

$$\mathbf{v}'_i = \sum_l w_{il}\ \mathbf{T}_l\ \mathbf{v}_i, \tag{1}$$

where $\mathbf{T}_l$ are some unknown time-varying link transforms. Moreover, we require the set of link correspondences L to have tree-structured connectivity defined by a function P that maps every link $l \in L$ to its unique parent $P(l)$. We also add an index $\omega \notin L$ and denote the link r whose parent is $P(r) = \omega$ the root node. Note that such a LBS description is the lowest common denominator of practically all articulated characters found in games.

3.2 Pipeline Process

Given the skinned input mesh, our method proceeds to estimate an articulated model as follows (refer to Fig. 3). In the *joint estimation* branch (lower part in Fig. 3) of our pipeline (see § 5.1), we first analyze the skinning weights and their link correspondences to segment the original geometry into an approximate set of body parts (f). From this segmentation, we then derive a filtered set of *oriented joint locations* (g) that consist of orientation vectors, and the joint's rotation centers that we place on an approximate medial axis representation of the fabrication mesh (h).

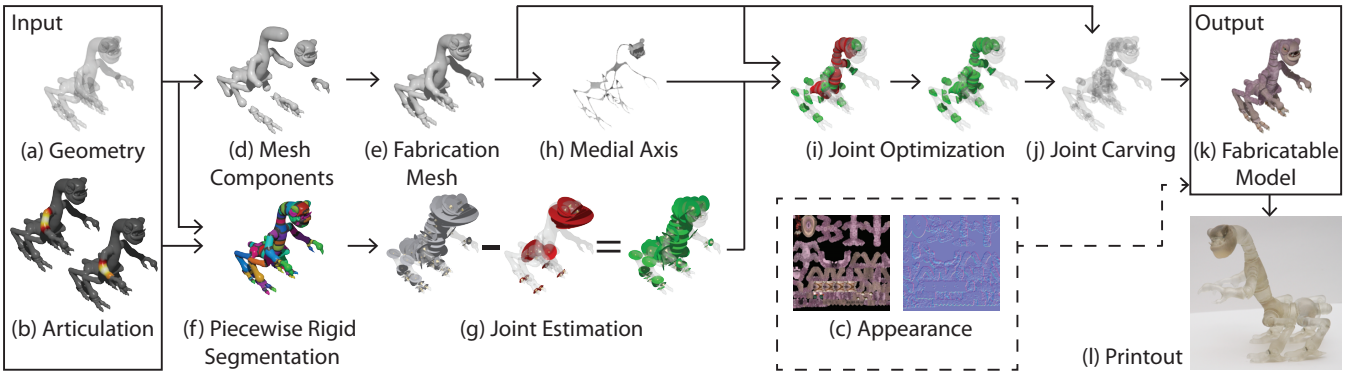


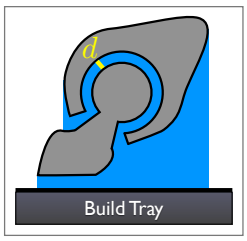
Figure 3: ***Pipeline Overview:*** *Given a skinned input mesh with (a) geometry, (b) skinning weights whose link correspondences are organized in a single rooted tree structure, and optional (c) diffuse texture and normal map, our approach estimates a (k) fabricatable 3D model as follows: (d) mesh components are identified, and (e) fused into a single, closed surface we call the fabrication mesh $\mathcal{F}$. Joints are computed by (f) estimating a rigid link segmentation from skinning weights, and (g) estimating proxy joint locations and filtering problematic joints. To optimize joint center placement, we use (h) an approximate medial axis representation of $\mathcal{F}$. (i) The parameters of volumetric joints with optional user-specified range constraints are optimized for strength and to avoid inter-joint collisions. (j) The joints are carved out of $\mathcal{F}$ using CSG operations. The final 3D printout (l) is a posable reproduction of the virtual articulated character.*

The fabrication mesh $\mathcal{F}$ (e) together with the articulation data (g) is then fed into our *joint optimization* procedure (i) where posable joints with maximal cross-sectional areas are being generated from corresponding oriented joint locations together with any user-specified range constraints. Pairwise collisions between generated joints are resolved while keeping the joints' rotation centers fixed (see § 5.4). Overall, our mapping tries to keep as much of the input articulation, while also keeping the model structurally strong. The final set of non-colliding, mechanical joints are then carved out of $\mathcal{F}$ using CSG (j) and we get a ready-to-print, structurally strong, articulated model (k) consisting of a set of piecewise-rigid parts that are jointed together with hinges, or balls and sockets. The models are statically posable using a joint friction design discussed in § 5.3.

Optionally, the joint carving can be performed on a colored, high resolution fabrication mesh whose geometric detail is computed by inverting normal mapping using the weighted least squares version of Nehab et al. [2005].

4 Manufacturing Considerations

Our posable output models are tailored to be fabricated on AM devices as single, assembled pieces. In contrast to subtractive manufacturing where material is "cut off," e.g., with a mill, additive manufacturing [Gibson et al. 2010] builds an input geometry layer-by-layer by, e.g., fusing plastic droplets or sintering metal powders via lasers. Unlike in 2D printing, heads of such "3D printers" move in a plane parallel to and above a horizontal plate called a *build tray*. Tiny particles of the *build material* are added, and, when a layer is finished, the tray is moved down a step, and the next layer is added.



To manufacture overhanging or assembled geometry like our mechanical joints, layered approaches use some kind of supporting structure as illustrated on the left in blue. After printing, this *support material* can either be blown (for powders), broken, or washed off. To ensure that the individual, assembled parts (in grey) are movable, and not fused during printing, we ensure a device-dependent *minimal distance* d (in yellow) between these pieces. Hence, we treat d as hard constraint when estimating our geometric joint models in § 5.2.

An important factor for manufacturability on AM devices is the models' structural strength because it puts a limit on the feasibility of desired output dimensions and largely affects the models' durability. If substructures are too fine, they either break off during fabrication, or when interacting with the final printouts.

When designing simple structures (e.g., trusses), civil engineers repeatedly identify their weakest link, and adjust its dimensions. Inspired by this basic analysis, we seek to increase the articulated models' overall strength by identifying and maximizing each of their mechanical joints' critical cross-sectional areas. We reject joints if their minimal cross-section falls below a technology imposed global, *critical area threshold* $A_{\min}$. While this heuristic does not ensure structural optimality, it allows us to formulate our hinges and ball-and-sockets using parametric, geometric models of joint strength (see § 5.2). Note that, because our virtual input characters might be nonphysical, e.g., cartoon characters, their corresponding fabrication meshes could themselves have critical sections below $A_{\min}$ as, e.g., in long and slim necks. However, we do not improve the structural strength of our input geometry.

5 Articulated Model Estimation

We now describe the estimation of oriented joint locations from the character's skin, and cast our hinges and ball-and-sockets as geometric models of joint strength that are then optimized while avoiding joint-joint collisions.

5.1 Estimating Rigid Parts and Joint Locations

To estimate oriented locations where mechanical joints are best placed (see Fig. 4), we exploit the link correspondence P encoded in the skinning weights w_{il} and ignore the character's rig. We observe that a segmentation of the character's input geometry (V, F) into piecewise rigid parts is naturally given by assigning each vertex i to the link l with maximal weight $\max_{i \in V} w_{il}$, as visualized in Fig. 4 (a) with a unique hue per link.

Most LBS descriptors lack information about skeletal joint motion (as implicitly encoded in the link transform parts $\mathbf{T}_l$ in Equation 1), and often include rig joint locations for the characters' rest pose only. Unlike skinning weights, rig joint locations are not a reli-

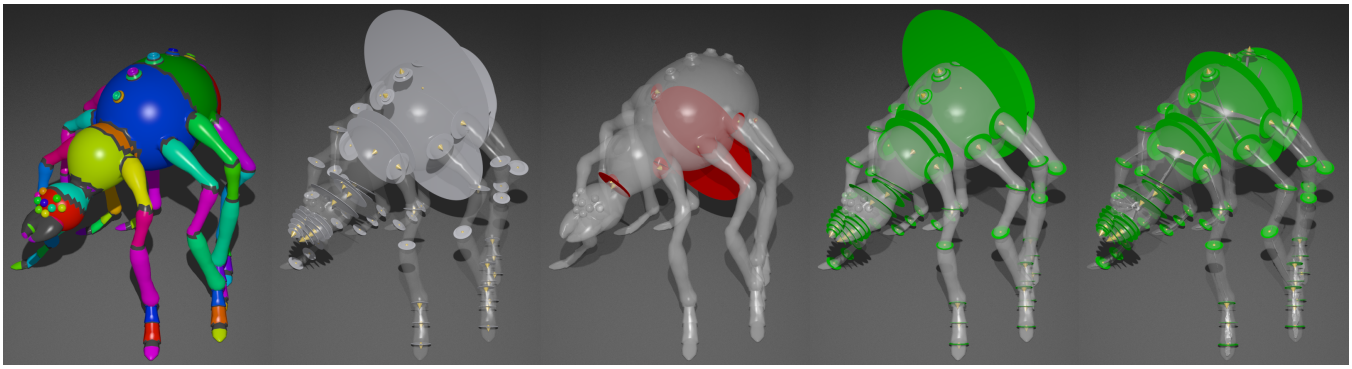
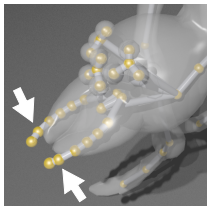


Figure 4: ***Estimating Articulation Behavior:*** *(a) Piecewise rigid segmentation using skinning weights. Faces whose vertices belong to different segments, are shown in black. (b) Transitions oriented from the root towards the leafs in the link connectivity P, (c) degenerate, and (d) filtered transitions. (e) Final set of joint locations on the scale axis transform of $\mathcal{F}$.*

able source for mechanical joint placement because they are non-physical, zero-dimensional points. Firstly, they are not guaranteed to be embedded in the character's geometry as demonstrated on the right for a spider's mandible. Secondly, rig joints typically do not fall together with actual rotation centers during animations as the cylinder example in Fig. 2 left illustrates. Hence, it is better to place joints at transitions of maximal link influence as shown in black in Fig. 4 (a). Such transitions are by default found in regions where the model bends most during animations and where joints are natural.



After segmentation, we approximate each transition with a plane (see Fig. 5) as illustrated in Fig. 4 (b) with gray disks. We first identify all unique edges in (V, F) whose end vertices j and k have maximal link influences $l_j = \arg\max_{l \in L} w_{jl}$ and $l_k = \arg\max_{l \in L} w_{kl}$ with $l_j \neq l_k$. Note that links l_j and l_k do not have to be direct neighbors in the tree-structured connectivity P even though they usually are. We then partition this set of *transition edges* with respect to matching ordered link-pairs (m, o)

$$\bigcup_{(m,o)} \{\{j,k\} \,|A(\{j,k\})\}\,, \tag{2}$$

with $A := ((l_j = m) \wedge (l_k = o)) \vee ((l_k = m) \wedge (l_j = o))$ and where link m is closer (or equal) to the root than o. Note that in rare cases where transitions (m, o) span over branches in P and where both links m and o have the same distance to the root, the link order is ambiguous. Transition (l_1, l_0) in Fig. 5 left provides an instance of such a case as both links l_1 and l_0 have r as a parent. To resolve this ambiguity, we randomly choose the link order (m, o). Alternatively, the user could specify it. For each edge $\{j, k\}$ in each *transition* (m, o) (see Fig. 5 right), we then compute a *transition point* $\mathbf{p}_{jk}$

$$\frac{w_{j,l_j}}{w_{j,l_j} + w_{k,l_k}} \mathbf{v}_j + \frac{w_{k,l_k}}{w_{k,l_k} + w_{j,l_j}} \mathbf{v}_k, \tag{3}$$

with normalized maximal weights w_{j,l_j} and w_{k,l_k}, and, finally, linearly approximate each transition by running Principal Component Analysis (PCA) on the set of corresponding transition points, resulting in a mean point $\mathbf{p}^{(m,o)}$ and principle components $\mathbf{e}_{\lambda_1}$, $\mathbf{e}_{\lambda_2}$, and $\mathbf{e}_{\lambda_3}$, sorted by their variances $\lambda_1 \leq \lambda_2 \leq \lambda_3$. We call the mean point *transition center* and the vector $\mathbf{n}^{(m,o)} = s\mathbf{e}_{\lambda_1}$, the *transition's orientation*. Next, we consistently orient planes (choosing the sign $s = \pm 1$) w.r.t. the hierarchical structure in P (from the root towards the leaves). While orientations do not affect the DOFs of individual mechanical joints in the posable output models, it allows us to pack the volumetric joints more closely, hence to keep more of the overall input articulation. We set s to 1 if more of the edge end vertices $\mathbf{v}_j$ (corresponding to the link m closer to the root, assuming $l_j = m$) are on the positive side of the transition plane ($(\mathbf{v}_j - \mathbf{p}^{(m,o)}) \cdot \mathbf{n}^{(m,o)} > 0$) than end vertices $\mathbf{v}_k$ on the plane's negative side ($(\mathbf{v}_k - \mathbf{p}^{(m,o)}) \cdot \mathbf{n}^{(m,o)} < 0$).

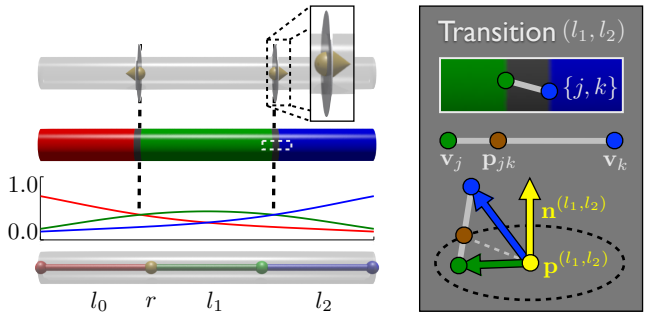


Figure 5: ***Estimating Transitions:*** *(left) Skinned cylinder with root r and three links (l_0 in red, l_1 in green, l_2 in blue) with their corresponding skinning weights (bottom). The link connectivity P is defined by $P(l_0) = r$, $P(l_1) = r$, $P(l_2) = l_1$, and $P(r) = \omega$. The two transitions (l_1, l_0) and (l_1, l_2) together with the final oriented transition planes pointing from the root towards the leaves in P (top). (right) A transition edge (in gray) with corresponding transition point (top) for transition (l_1, l_2). (right) From the transition points, and their edges' end vertices (in blue and green), we compute the transition's center and orientation (in yellow, bottom).*

Taking a closer look at the estimated transitions (see Fig. 6), we observe that their corresponding transition points do not always span a closed loop on the input geometry, as illustrated in Fig. 6 and 4 (c) with red disks. Because it is unclear how a mechanical joint should be placed on a transition that, e.g., only covers half of the geometry, we filter out such *degenerate transitions*. We find that a good measure for degeneracy is given by the ratio of the largest- and mid-eigenvalue of the 3x3 PCA covariance matrix at (m, o) because it clearly discriminates between cases where transition points are close to circularly distributed (green disks in Fig. 6) and the degenerate cases. If the largest variance λ_3 is at least a factor f larger than the mid-variance λ_2, we reject the transition. This leaves us with the set of transitions shown in Fig. 4 (d).

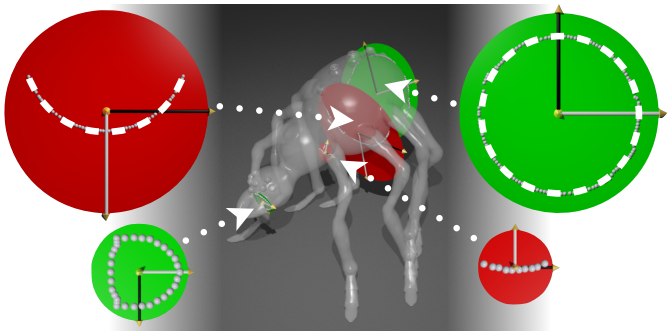
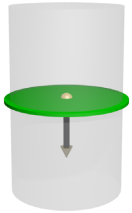


Figure 6: ***Filtering Transitions:*** *While for valid joints, transition points (gray) span a closed loop on the input geometry (green disks). However, for a subset of transitions (red disks), they only cover a partial loop on the geometry, indicating that the two corresponding body parts are semi-rigidly connected. Because it is unclear how a mechanical joint should be placed for such degenerate transitions, we filter them out.*

Because it is unclear from the articulation data where to best place joint centers on the transitions, we set the centers to the closest intersection $\mathbf{c}^{(m,o)}$ of $(\mathbf{p}^{(m,o)}, \mathbf{n}^{(m,o)})$ with an approximate medial axis representation of the fabrication mesh $\mathcal{F}$. Because the medial axis transform [Blum 1967] is unstable and leads to many unintuitive branches, we use the recent scale axis transform [Miklos et al. 2010] instead. Placing joint centers on the scale axis is reasonable because it allows to maximize the mechanical joints' sizes, hence, to leverage their structural strength. Furthermore, this choice guarantees that the joints' center is always in the interior of $\mathcal{F}$. The final set of oriented joint locations $(\mathbf{c}, \mathbf{n})$ is shown in Fig. 4 (e).

5.2 Optimizing Parametric Joints for Strength



Given an oriented joint location $(\mathbf{c}, \mathbf{n})$, as illustrated on the left with a cylinder with a single mid-transition, we now estimate mechanical joints. To this end, we cast our hinge and ball-and-socket designs as parametric, geometric models of joint strength (see Fig. 7 left). To minimize interference of the joints with the character's overall appearance, we limit their parameters so that the sockets for both designs are guaranteed to be embedded in the maximum inscribed sphere of radius $r_{\max}$ in the fabrication mesh $\mathcal{F}$, at the joint's rotation center $\mathbf{c}$ (see dotted, black circles in Fig. 7). Furthermore, we keep a minimal distance d between the joint parts to prevent their fusion during manufacturing.

When designing structures, civil engineers repeatedly analyze the stress distribution within the structures' bodies under a set of typical loading scenarios (e.g., [Beer et al. 2011]). A simple view is that the average *stress* across a given *cross-section* A is given by the force per area $\sigma = F/A$, where F is the residual load. If a local stress level is too high, a structure could break, hence, they adjust the design's dimensions in that particular region, thereby increasing the corresponding *critical* area. In the same spirit, we identify a total of three critical cross-sectional areas for each of our designs (see Fig. 7 right) and maximize each joint's minimal area. While these critical areas are parameterized with only two parameters for our ball-and-sockets (the socket's radius r and a height parameter h, see Fig. 7 top row, left), we need three parameters for our hinges: The outer and inner radii R and r, and the width b, limiting the hinge's toroid (see Fig. 7 bottom row, left). This leads to the following two constrained max-min optimization problems.

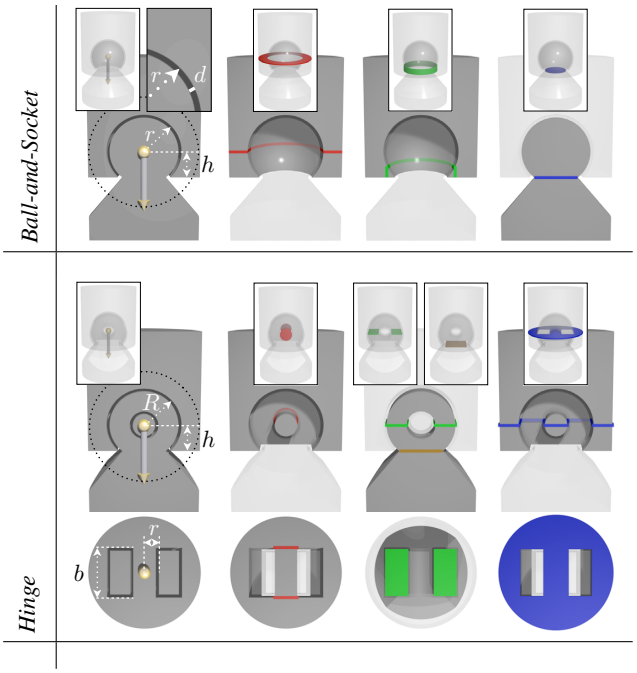
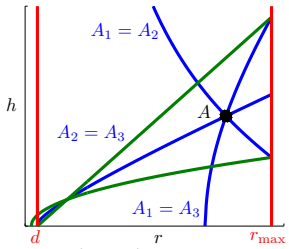


Figure 7: ***Critical Cross-Sectional Areas:*** *(top) Our ball-and-socket design with its critical areas A_1 (red, circle of radius r_{max} with centric hole of radius r), A_2 (green, open cylinder of radius $r-d$ and height $h-\sqrt{r^2-(r-d)^2}$), and A_3 (blue, circle of radius $\sqrt{(r-d)^2-h^2}$). (bottom) For our hinge design, we get A_1 (red, twice the area of circle with radius $r-d$, assuming this section to break in double-shear [Beer et al. 2011]), A_2 (green, twice the rectangular area with sides $b-2d$ and $R-d-r$), and A_3 (blue, circle with radius r_{max} reduced by twice the rectangular area with sides b and $R-(r-d)$). Area A_4 (brown) is non-critical because for all feasible hinges, there is a h so that $A_4 \geq A_2$. In practice, we choose h so that areas A_2 and A_4 are equal.*

Ball-And-Socket Joint: For our *ball-and-socket design*, we get

$$\max_{\{r,h\}} \min_{i \in I} A_i(r,h), \qquad (4)$$



with $I = \{1,2,3\}$ and constraints $r_{\max} > r > d$ and $r - d > h > \sqrt{r^2-(r-d)^2}$ limiting the joint's feasibility as shown on the right in red and green, respectively. Note how the curves corresponding to equal areas (in blue) meet at a single point A. For almost all pairs $(d, r_{\max})$, our max-min problem leads to three equal critical areas. If the joint is infeasible or its minimal critical area is below the global threshold $A_{\min}$, we reject it.

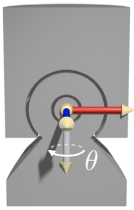
Hinge Joint: Similarly, we get

$$\max_{\{R,r,b\}} \min_{i \in I} A_i(R,r,b), \qquad (5)$$

with $I = \{1,2,3\}$ and constrained by $r > d$, $R > d + r$, $r_{\max}^2 > \left(\frac{b}{2}\right)^2 + R^2$, and $b > 2d$ for our *hinge design*.

Note, however, that the ranges for our current designs are limited in directions perpendicular to the joint's orientation (compare with Fig. 7 left). While rotational joint motion is too restrictive for our current hinges, joint motion around axis $\mathbf{n}$ is unrestricted for our ball-and-sockets. These spherical joints are there-

fore well-suited for common joints found in hips and spines. For elbow, knee, or shoulder joints, however, they are unfit. Because it is unclear how to estimate joint types, ranges, and the hinges' rotation axes from the character's skin, we give the user the option to specify them. Because general ranges are not rotation-invariant w.r.t. angle-axis $(\theta, \mathbf{n})$, we disambiguate by introducing a right-handed, orthogonal *joint frame* $[\mathbf{a}, \mathbf{n}, \mathbf{f}]$ whose *forward* axis $\mathbf{f}$ (red arrow on the left) is aligned with the direction where θ is zero. Note how axis $\mathbf{a}$ (in blue) falls together with our hinge's rotation axis.



User-Intervention: This frame is uniquely defined by our estimated joint locations, up to the axis' $\mathbf{a}$ rotation angle w.r.t. the joint's orientation that we let the user choose. Ranges can then be specified by direction-dependent *opening angles* $\phi(\theta)$ for our ball-and-sockets, and forward (γ_f) and backward (γ_b) *swing angles* for our hinges (see Fig. 8 left).

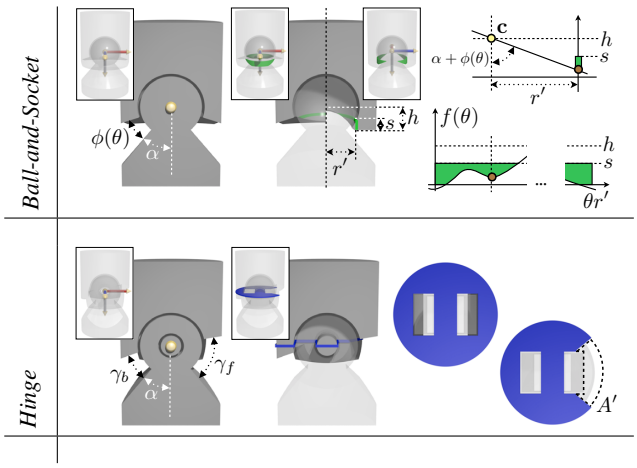


Figure 8: ***Joint Ranges:*** *(top) Range constraints for our ball-and-sockets may reduce open cylinder area A_2 (green) of radius r' and height s. The "unrolled" cylinder area (see graph in the lower right) is reduced by the area under $f(\theta)$ that overlaps with range $[0, s]$. Value f at a θ (brown point) is given by the intersection of line through joint center $\mathbf{c}$ and slope $\tan(\alpha + \phi(\theta))^{-1}$, with the infinite cylinder of radius r' (see upper right, note that $\cos\alpha = \frac{h}{r'}$). (bottom) Forward and backward constraints for our hinges may reduce critical area A_4 by A' each, as illustrated with a swing angle γ_f that leads to a combined angle $\alpha + \gamma_f$ larger than $90°$ (with $\cos\alpha = \frac{h}{R-d}$).*

Range Constraints: These *range constraints* may reduce critical areas of our joint designs as illustrated in Fig. 8 right. For our hinges (bottom row), a swing angle that is – when combined with α – larger than $90°$, reduces section A_3 by an amount A'. This reduction can be expressed in closed form, parametrized by the hinge's set of parameters. To incorporate the range constraint $\phi(\theta)$ into our ball-and-socket design (top row in Fig. 8), we reduce the cylindric area A_2 with circumference $2\pi r'$ ($r' = r - d$) by

$$\int_0^{2\pi} \min\left(s, \max\left(0, f(\theta)\right)\right) \theta r' d\theta, \qquad (6)$$

with cylinder height $s = \sqrt{r^2 - r'^2}$ and $f(\theta) = h - \frac{r'}{\tan(\alpha+\phi(\theta))}$. A similar derivation leads to a reduction of area A_1 in cases where the sum of the maximal opening angle and α is larger than $90°$.

Note that we recompute these critical areas with their reductions in each iteration of our joint optimizations, and that our max-min formulations balance these areas up to equality as long as the constraints allow it. Infeasible designs, such as a socket that cannot hold its ball, are caught by our feasibility constraints. Without user-intervention, we can automatically generate articulated models with spherical *default joints* with constant, global constraint $\phi(\theta) = \beta$. Our geometric formulations, however, are only approximate models for joint strength and optimality w.r.t. structural strength is not guaranteed. Nevertheless, we avoid weak joints by maximizing their minimal critical cross-section and rejecting them if this section has a value below the global threshold $A_{\min}$. Also, while our two basic joint types lead to output models with sufficient DOFs, our recipe of identifying critical sections and maximizing their minima is general and applies to other joint designs also.

5.3 Fabricating Posable Joints with Friction

From the joints' blue prints (see Fig. 7 and 8 left) together with device-dependent manufacturing, user-provided range, and estimated joint parameters, we then generate an implicit CSG representation of the volume (in green in Fig. 9) that we have to remove from fabrication mesh $\mathcal{F}$ to introduce a joint at its estimated location. We call this volume *joint hull*. After polygonizing these hulls, we carve them out of $\mathcal{F}$ with mesh-boolean difference operations (see Fig. 9 right), resulting in fabricatable output models with desired kinematics. These models, however, are unlikely to retain a pose once placed into it, and are more like a printed "rag doll." To overcome this limitation, we fabricate small bump spheres of radius r_b onto the positive joint parts similar to [Grey 1999; Wai 2006]. To prevent fusion of movable parts during manufacturing, we extent their ideas by subtracting spheres with same centers but extended radius $r_b + d$ from the negative joint parts also, as illustrated in the top, right corner in Fig. 9. This additional friction mechanism results in posable joints with continuous position control. While these *friction bumps* could potentially stick out of $\mathcal{F}$ after joint carving, we did not observe such cases when estimating our demonstration models. To guarantee embeddedness, we could reduce radii $r_{\max}$ by r_b or, alternatively, invert the bumps and add them to the negative joint parts instead.

5.4 Avoiding Joint-Joint Collisions

As of now, we can successfully turn simple skins into posable output models, consisting of a set of jointed, rigid pieces that we can print assembled. For sophisticated input skins, however, estimated joint locations are often in close proximity to one another, and, as

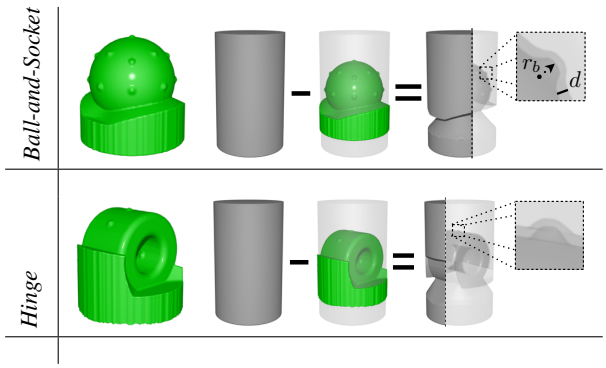


Figure 9: ***Frictional Joint Designs*** *based on adding small calibrated bumps. (top) Ball-and-socket joint hull with friction bumps on the ball part and (bottom) hinge joint hull with bumps on the toroidal part. Printed articulated models can then retain their pose.*

aforementioned and illustrated in Fig. 2 right, corresponding joint hulls are likely to collide when we maximize the individual joints' sizes. Such overlaps may lead to broken joints, as a closer look at an example of two colliding hulls unveils: if, e.g., a hull volume of one joint contains the part of another spherical joint's socket that prevents its corresponding ball from popping out, we get two disassembled pieces in our output. Hence, we resolve such joint-joint collisions before carving their hulls out of the fabrication mesh $\mathcal{F}$.

In a first naive approach, we could simply remove individual joints, until there are no further hull collisions left. However, while this strategy guarantees functioning joints in our output models, it is not optimal, because we would reject far more of the "fabricatable" input articulation than necessary. A second approach would act directly on what causes the collisions in the first place: the proximity between estimated joint locations. By moving these locations, we could "fit" more joints in $\mathcal{F}$. However, because we set the joints' rotation centers to these locations, this second strategy would significantly change the semantics encoded in our input articulation (if locations were moved away from their corresponding transitions). In the following, we describe our collision resolution procedure that tries to keep as much of the input articulation as possible while avoiding weak joints and keeping their rotation centers fixed. See Fig. 10 and the accompanying video for illustrations.

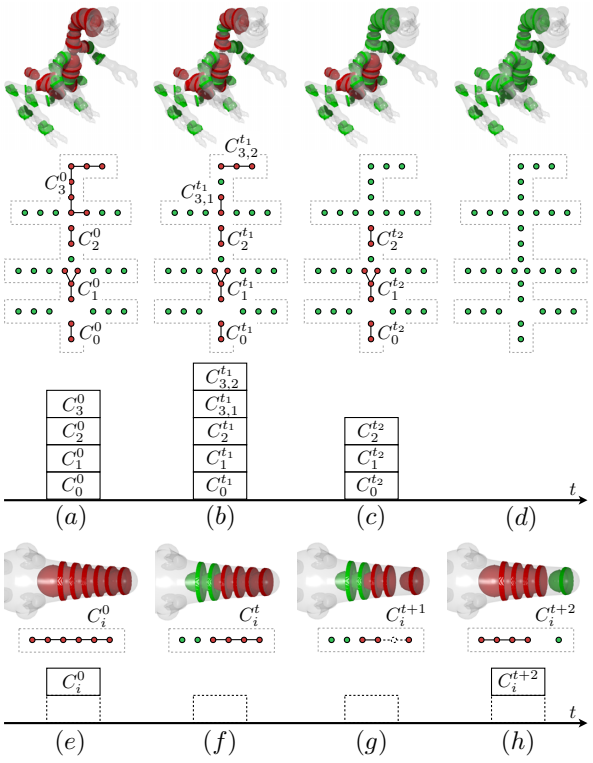
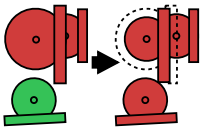


Figure 10: ***Resolving Collisions:*** *Colliding joints are shown in red, non-colliding joints in green. For top (a-d) and bottom row (e-h), we have joint hulls on top, corresponding collision graph, and stack in the middle and at the bottom. (a) Initial collision groups for a full character, (b) group split after a resolution, (c) completion of a collision group, and (d) final set of non-colliding joint hulls that we then carve out of $\mathcal{F}$. (e) Initial collision group for a character's tail, (f) a joint gets infeasible (A_{min} too small), (g) exclusion of a joint, (h) updated joint hulls and collisions after a group reset.*

To initialize our resolution process, we proceed as described in § 5.1, 5.2, and 5.3. We compute the radius $r_{\max}$ of the maximum inscribed sphere, then optimize a parametric joint model consistent with any user-specified ranges at each estimated location, resulting in a set of joint hulls. Next, we compute all pairwise collisions between these hulls that we inflate by half the distance d, to guarantee a minimal offset between individual joints also. (Note that when we speak of collisions in the following we refer to collisions between such *inflated joint hulls*). To coordinate further processing, we then abstract joint hulls with nodes and pairwise collisions with undirected edges of what we call a *collision graph*. Thereafter, we extract all connected components of this graph with orders larger than one, and push this *collision groups* onto a *collision stack*. Refer to Fig. 10 (a), where we use the notation C_i^t to uniquely identify each group i at time step t of our resolution.

As long as there are groups on this stack, we pop the topmost and repeatedly reduce the radius $r_{\max}$ for the joint with largest minimal cross section, as it is currently the strongest within this group. We then reestimate its optimal parameters, and check for collisions with its updated joint hull. We stop when either a collision (or several) got resolved, a joint gets infeasible (e.g., a joint's minimal critical area gets smaller than $A_{\min}$), or a joint hull is colliding with a hull outside of its collision group. While such *outside collisions* are rare in practice, it is crucial to check for them, as the example of three spherical joint hulls in the inset figure on the left illustrates. When we reduce the size of the "strongest" of the upper pair of colliding joints, we introduce a second collision with a "node" outside of that group.



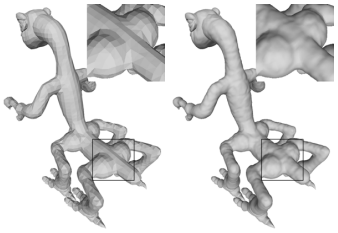
If collisions got resolved, we are either done (no more collisions within this group) and continue (see Fig. 10 (c)), or split the collision group into subgroups, if necessary, and push those onto the stack. See Fig. 10 (b) for an illustration, where we use $C_{x,j}^t$ to denote the subgroup j with previous *group correspondence history* x. If no split is required (single group), we simply push back C_x^t, without the resolved "edges" and "nodes". However, if a joint becomes infeasible or a member collides with a joint outside of its collision group, this group is unresolvable without excluding a joint. (Note that while we could add outside collisions to groups or merge groups of the involved members, such "additions" or "merges" may lead to cyclic behavior in our resolution process. Hence, we exclude a joint instead thereby guaranteeing convergence.) We observe that a good candidate for exclusion is given by the member of the current group that was "weakest" after initialization (smallest $A_{\min}$). While this heuristic leads to pleasing output models in practice, this to-be-excluded joint could also be chosen by the user. After an exclusion, we pop all descendants of the original collision group (all groups that have first index k in their correspondence history, if k is the original group's index after initialization), and push the original collision group (k) with reset radii $r_{\max}$ and without the excluded joint back onto the stack. Such a reset is necessary because an exclusion of a joint might make previous reductions of joint sizes unnecessary.

Note that our collision resolution process performs evenly well on any other parametric joint designs (other than our hinges and ball-and-sockets from § 5.2) as our collision handling is evaluated on arbitrary hulls, with the only requirement that the joints have to have a single rotation center. Because joints can only get smaller and we exclude a joint if a member gets infeasible or collides with an outside joint, our collision process converges.

6 Results

We have created and printed a total of six models based on five skinned characters generated by the SPORE Content Creator ("Grumpy" in Fig. 1, "Chicks" and "Dinofrog" in Fig. 11,

"Cristal Frog" and "Lippy" in Fig. 12), and a realistic human hand model that we rigged and skinned in Maya (see Fig. 13). Our five SPORE examples include diffuse and normal maps, and joints were carved out of their colored fabrication meshes, whose geometric detail we computed by inverting normal mapping [Nehab et al. 2005]. This inversion leads to significant quality improvements in $\mathcal{F}$, hence, also in our printouts, as illustrated on the right with a comparison of input and reconstructed geometry for our "Grumpy" character. All of our articulated output models were printed with an Objet Connex 500 printer that has a resolution of 600 DPI on the horizontal x and y axis, and 1600 DPI on the vertical z axis. We used three of Objet's hard, plastic-like materials called "VeroBlack" ("Lippy" and "Cristalfrog"), "VeroClear" ("Grumpy," "Chicks," and "Dinofrog"), and "ABS-like digital material" (hand model). While "VeroClear" is transparent and the embedded joints, therefore, visible, the ABS-like material is the structurally strongest (e.g., LEGO is made out of ABS). Objet's support material is gel-like and can be removed with a water-jet.



To identify the minimal offset d to ensure jointed parts to be movable, and the critical area threshold $A_{\min}$ to avoid weak mechanical joints, we estimated hinges and ball-and-sockets for a single-transition cylinder (see Fig. 9 right) with varying radius and for different offsets d, and then printed them with the three printer materials: beyond offsets of 0.3 mm, parts started fusing and the support material could not be water-jetted or "broken out" any longer, and joints with minimal critical areas smaller than 10 mm^2 for "VeroClear" and "VeroBlack", and 3 mm^2 for the ABS-like material, started to get brittle. With a similar empirical experiment, we identified a friction bump radius r_b of 0.7 mm. Note that this bump radius is larger than the minimal distance d.

Prior to our articulation estimation, we scaled our input to target sizes (in direction normal to the ground plane shown in gray in Fig. 1, 11, 12, 13) of our output models: 150 mm for "Grumpy" and our hand model, 85 mm for "Chicks," and 100 mm for "Lippy," "Dinofrog," and "Cristalfrog". To filter degenerate transitions, we used factors $f \in [3.5, 4.0]$. Generally, very little user-intervention is needed. E.g., for "Grumpy," the user-intervention was restricted to switching 10 joints to hinge type and specifying three angles each (forward and backward swing angles, and rotation angle around the joint's estimated orientation axis). In addition, we specified spherical range constraints for three neck joints (with again, three user-specified angles each, because we use elliptical opening angles $\phi(\theta) = \phi_a \sin\theta + \phi_b \cos\theta$). All other joints are defaults with global, rotation-invariant range β of a few degrees. With our unoptimized implementation that uses an implicit, extended, regular-grid-based marching cubes approach, it takes approximately 5.5 hours to process "Grumpy," which is still a fraction of the needed manufacturing time of 18 hours. The time required for processing highly depends on the number of collisions that have to be resolved prior to joint carving. While our SPORE examples had many collisions to resolve, our hand model only had a single collision between two neighboring knuckle joints (overall processing time under 10 min).

7 Conclusions and Discussion

We have devised a method to generate fabricatable characters from skinned input meshes, e.g., suitable for personalized posable toys. While we are able to generate characters with spherical default joints fully automatically, we allow users to specify joint types and ranges for joints where defaults are not as natural. Note that input

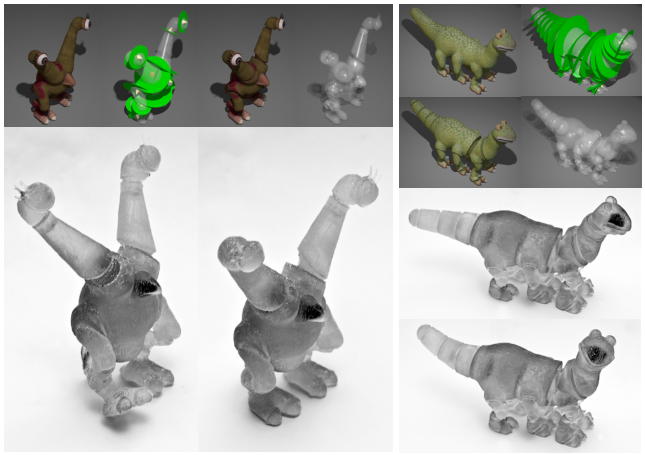


Figure 11: ***"Chicks" and "Dinofrog"***

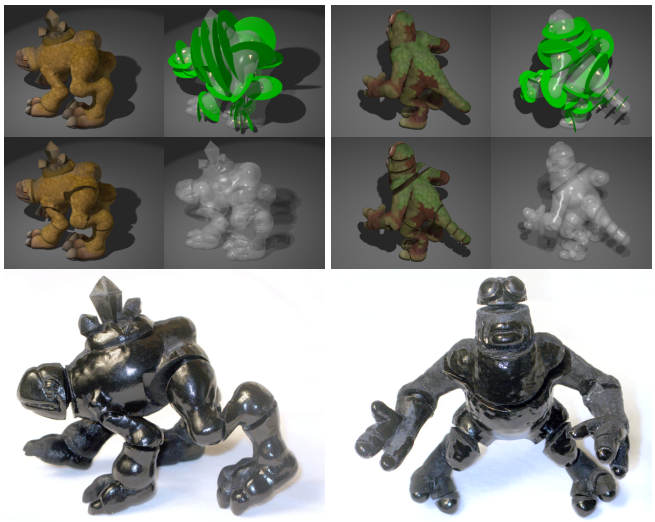


Figure 12: ***"Cristal Frog" and "Lippy"***

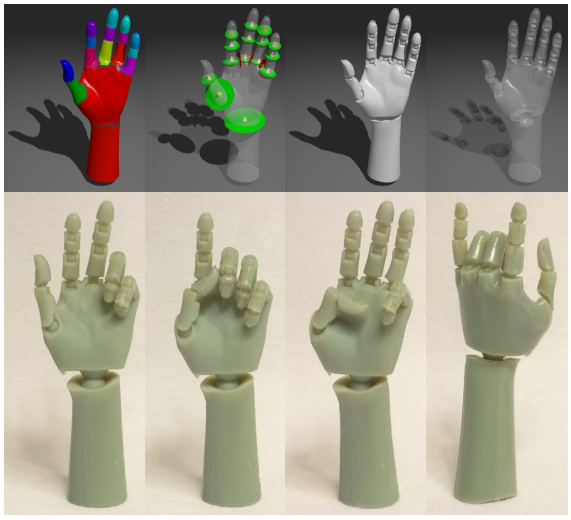


Figure 13: ***"Hand"***

skins have transitions where joints are expected, because transitions between joint influences are naturally at places where the model's geometry bends the most during animations. However, while we could always have the user remove unwanted transitions and corresponding joints if there are too many, our system is not able to estimate joints where there is no input data. In the future, we expect that our method and its successors will enable a fully "automatic 3D print button" for characters.

There are several remaining challenges. Current 3D printers introduce many limitations on what we can print. Although our system fully supports colored characters, we were not able to print posable articulated output models in full color. Furthermore, while we avoid weak joints by optimizing parameters of our geometric approximate models of joint strength, our hinge and ball-and-socket designs are not modeling structural strength to a level of accuracy where our system could be fed with a set of measured material parameters to estimate *structurally optimal* joints. As aforementioned, our input skins could also include fine geometric detail with cross sections smaller than $A_{\min}$, or even parts that are completely disconnected from the model's main body, or overlap in the character's rest pose. This would require to either significantly changing the input geometry (locally inflate geometry, adding artificial connectors, etc.) or rejecting those parts completely. Also, our articulated outputs can be understood as first order, piecewise linear approximate reproductions of the virtual input articulation. Complete piecewise continuous reproductions that include a deformable skin, are left as future work.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments, Kalyan Sunkavalli, Christian Regg, Maurizio Nitti, Nicolas Bonnet, Samuel Muff, and James Tompkin for their help with the video, figures, Maya (hand model), and printing, respectively. We also wish to thank Robert Wood, Radhika Nagpal, the Harvard Wyss Institute for Biologically Inspired Engineering, Joe Marks, and Disney Research Boston for letting us use their Objet printers. We also wish to thank Miloš Hašan, the VCG group at Harvard and the CG group at TU Berlin for insightful discussions. This work was partially supported by NSF grant IIS-1116619. Doug James acknowledges support from Pixar and a fellowship from the John Simon Guggenheim Memorial Foundation.

References

ABBAT, J.-P., 1993. Articulated doll joint. U.S. Patent 5257873.

BARAN, I., AND POPOVIĆ, J. 2007. Automatic rigging and animation of 3D characters. *ACM Trans. Graph. 26*, 3 (July), 72:1–72:8.

BEER, F., JOHNSTON, E. R., DEWOLF, J. T., AND MAZUREK, D. F. 2011. *Mechanics of Materials*, 6 ed. McGraw-Hill.

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. *ACM Trans. Graph. 29* (July), 63:1–63:10.

BLUM, H. 1967. A Transformation for Extracting New Descriptors of Shape. In *Models for the Perception of Speech and Visual Form*, W. Wathen-Dunn, Ed. MIT Press, Cambridge, 362–380.

DONG, Y., WANG, J., PELLACINI, F., TONG, X., AND GUO, B. 2010. Fabricating spatially-varying subsurface scattering. *ACM Trans. Graph. 29* (July), 62:1–62:10.

FERRE, R., 2000. Form of articulated structures for dolls or puppet bodies. U.S. Patent 6033284.

GIBSON, I., ROSEN, D. W., AND STUCKER, B. 2010. *Additive Manufacturing Tehnologies*. Springer.

GREY, M. J., 1999. Construction system. U.S. Patent 5897417.

HAŠAN, M., FUCHS, M., MATUSIK, W., PFISTER, H., AND RUSINKIEWICZ, S. 2010. Physical reproduction of materials with specified subsurface scattering. *ACM Trans. Graph. 29* (July), 61:1–61:10.

HECKER, C., RAABE, B., ENSLOW, R. W., DEWEESE, J., MAYNARD, J., AND VAN PROOIJEN, K. 2008. Real-time motion retargeting to highly varied user-created morphologies. *ACM Trans. Graph. 27*, 3 (Aug.), 27:1–27:11.

JAMES, D. L., AND TWIGG, C. D. 2005. Skinning mesh animations. *ACM Trans. Graph. 24*, 3 (Aug.), 399–407.

KAVAN, L., COLLINS, S., ZARA, J., AND O'SULLIVAN, C. 2008. Geometric skinning with approximate dual quaternion blending. *ACM Trans. Graph. 27*, 4 (Oct.), 105:1–105:23.

KRY, P. G., JAMES, D. L., AND PAI, D. K. 2002. EigenSkin: Real Time Large Deformation Character Skinning in Hardware. In *ACM SIGGRAPH SCA*, 153–160.

LAU, M., OHGAWARA, A., MITANI, J., AND IGARASHI, T. 2011. Converting 3D furniture models to fabricatable parts and connectors. *ACM Trans. Graph. 30*, 4 (Aug.), 85:1–85:6.

LEWIS, J. P., CORDNER, M., AND FONG, N. 2000. Pose space deformations: A unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of ACM SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, 165–172.

MATUSIK, W., AJDIN, B., GU, J., LAWRENCE, J., LENSCH, H. P. A., PELLACINI, F., AND RUSINKIEWICZ, S. 2009. Printing spatially-varying reflectance. *ACM Trans. Graph. 28* (December), 128:1–128:9.

MIKLOS, B., GIESEN, J., AND PAULY, M. 2010. Discrete scale axis representations for 3D geometry. *ACM Trans. Graph. 29* (July), 101:1–101:10.

MOHR, A., AND GLEICHER, M. 2003. Building efficient, accurate character skins from examples. *ACM Trans. Graph. 22*, 3 (July), 562–568.

NEHAB, D., RUSINKIEWICZ, S., DAVIS, J., AND RAMAMOORTHI, R. 2005. Efficiently combining positions and normals for precise 3D geometry. *ACM Trans. Graph. 24* (July), 536–543.

SMITH, J., HODGINS, J., OPPENHEIM, I., AND WITKIN, A. 2002. Creating models of truss structures with optimization. *ACM Trans. Graph. 21* (July), 295–301.

WAI, F. C. A., 2006. Frictional joint for toys. U.S. Patent 7566256.

WANG, R. Y., PULLI, K., AND POPOVIĆ, J. 2007. Real-time enveloping with rotational regression. *ACM Trans. Graph. 26*, 3 (July), 73:1–73:9.

WEYRICH, T., PEERS, P., MATUSIK, W., AND RUSINKIEWICZ, S. 2009. Fabricating microgeometry for custom surface reflectance. *ACM Trans. Graph. 28* (July), 32:1–32:6.

WHITING, E., OCHSENDORF, J., AND DURAND, F. 2009. Procedural modeling of structurally-sound masonry buildings. *ACM Trans. Graph. 28* (December), 112:1–112:9.