

3D-Printing of Non-Assembly, Articulated Models

Jacques Calì Dan A. Calian Cristina Amati Rebecca Kleinberger Anthony Steed Jan Kautz Tim Weyrich
University College London



Figure 1: Starting from a 3D mesh (left), our system allows to intuitively add 3D-printable joints (center) that, when 3D-printed, create a functional, posable model with joints that exhibit internal friction. The model leaves the printer ready to use; no manual assembly is required.

Abstract

Additive manufacturing (3D printing) is commonly used to produce physical models for a wide variety of applications, from archaeology to design. While static models are directly supported, it is desirable to also be able to print models with functional articulations, such as a hand with joints and knuckles, without the need for manual assembly of joint components. Apart from having to address limitations inherent to the printing process, this poses a particular challenge for articulated models that should be posable: to allow the model to hold a pose, joints need to exhibit internal friction to withstand gravity, without their parts fusing during 3D printing. This has not been possible with previous printable joint designs. In this paper, we propose a method for converting 3D models into printable, functional, non-assembly models with internal friction. To this end, we have designed an intuitive workflow that takes an appropriately rigged 3D model, automatically fits novel 3D-printable and posable joints, and provides an interface for specifying rotational constraints. We show a number of results for different articulated models, demonstrating the effectiveness of our method.

Keywords: 3D Printing, Articulated Models, Mechanical Joints

Links: [DL](#) [PDF](#)

1 Introduction

Additive manufacturing (3D printing) is commonly used to produce physical models for a wide variety of applications, from archaeology to industrial prototyping and design. While having been

available for more than a decade, the technology recently developed additional momentum. With 3D printers dropping in price and with 3D printing services becoming available to a wider public, many additional use cases arise. For instance, rapid prototyping is now even used to print custom toys and figurines: there are commercial services that fabricate figurines from multiple-choice designs of puppets [MakieLab 2012] and robot toys [Kodama Studios 2012], or even based on models from World of Warcraft and other online games [FigurePrints LLC 2012].

Our work is inspired by—but not limited to—such figurine production. Observing the existing tools for fabrication of articulated models, it becomes apparent that they only allow for the fabrication of *static* models, or employ prefabricated parts to provide a limited degree of articulation of a *manually assembled* model. This evidently does not fully exploit the flexibility of modern 3D printing processes. In this paper, we show how to equip 3D models with *custom articulation*, with minimal user input, leading to designs that can directly be 3D-printed, without the need for manual part assembly.

3D printing of functional models has been researched as long as 3D printers exist. Simple sets of gears have been demonstrated early on, and by now they even attracted a hobbyist community [Thingiverse 2012]. Custom deformable objects have been printed using using a multi-material printer [Bickel et al. 2010]. Closely related to our work, several 3D-printable joints have been designed to be used in a robotic hand [Won et al. 2000; Mavroidis et al. 2001; Laurentis and Mavroidis 2004]; however, these joints require manual assembly and were manually designed for a specific model. Designing a different jointed model would require considerable amount of work. Also, these joints do not exhibit internal friction that would be required for a *posable* model.

In contrast, our work aims for functional, posable models that do not require assembly of any kind, as this would take away some of the advantage of using rapid prototyping techniques. This is challenging, as joints need to provide friction while maintaining certain tolerances in the printing process. Furthermore, we strive to make the final printed models aesthetically pleasing. To this end, we propose an intuitive workflow that takes as input a static 3D model. The user rigs this model for 3D printing using a standard rigging procedure [Autodesk 2012]. We then craft and fit newly developed 3D-printable joints in accordance with the provided rigging while taking into consideration the rotational constraints as specified by

the user, as well as the constraints imposed by the printing process. The main contributions of this paper are:

- Derivation of a single, versatile physical joint design that allows to incorporate adaptations to overcome limitations caused by the printing process (limited accuracy, treatment of excess or support material) and still creates controlled friction to support static poses.
- An interactive joint fitting pipeline that makes use of a simple rigging structure and outputs 3D-printable models requiring no assembly
- A set of automatisms that support this pipeline by sculpting joints and geometry according to the specified rotation constraints while otherwise allowing for free rotation of the joint.

We demonstrate our method on a variety of different models and for two different 3D printing processes. Figure 1 shows an example of taking an input mesh, fitting joints to it, and the final fabricated model.

2 Background

Computer-aided design (CAD) and computer-aided manufacture (CAM) are one of the most well-developed areas of computing, with dozens of commercial packages that target different areas of manufacture [Lee 1999; Alavala 2010; Leondes 2003]. Although it is a simplification, CAD programs tend to highlight the specifics of design creation with the aim of creating plans or renderings that can then be interpreted by skilled engineers or craftsmen, or are an end in themselves. There are very many machine or hand-controlled manufacturing processes [Thompson 2007], but for those that can be computer-controlled CAM design packages can turn designs into machining instructions. Computer controlled milling is the most obvious CAM example.

In the past decade new types of additive manufacturing process have become available. These offer new opportunities to designers and engineers because of their unique capabilities compared to previous manufacturing processes. Additive manufacturing provides the ability to design and build structures that would be difficult to construct with other processes. For example, certain additive manufacturing processes can build layered solids that would be impossible to mill, because the interior would not be reachable by the machining head. Finally, most software that drives additive manufacturing machines uses a standard file format that is easily generated by most CAD packages, thus alleviating the need for specialist software.

The term additive manufacturing covers a few different construction techniques [Dimitrov et al. 2006; Gibson et al. 2009]. Stereolithography (SLA) uses an ultraviolet curable liquid and an ultraviolet laser to solidify layers of the liquid into the desired shape. The layers of liquid are added gradually until the whole model is built up. With the Fused Deposition Model (FDM) technique, a thin filament of hot thermoplastic material is extruded from a nozzle onto a platform. The material solidifies instantly thus creating stratified structures. Both SLA and FDM have problems building structures with overhangs or interlocking parts, as the structure needs support as it is built. Selective Laser Sintering (SLS) makes use of granular material that can be either thermoplastics, metal or ceramic. This process works by sintering the powder layer by layer to form the solid structure using a guided laser. The parts of the powder which are untouched by the laser can act as a support material to support overhanging sections or loose components making it possible to print joint structures. FDM is a relatively cheap technology that has attracted a vibrant hobbyist scene [Stemp-Morlock 2010] but

the resolution of printing is coarser than SLA and SLS. We will use SLA and SLS in our examples as described in Section 7.

The relationship between computer graphics and additive manufacturing has developed in two distinct ways. First, additive manufacture has been considered to be a new output device for computer-generated designs [McMains et al. 2005]. A variety of novel design processes have targeted 3D printed output: from computer-generated sculptures [Xin et al. 2011; Mitra and Pauly 2009], through 3D puzzles [Lo et al. 2009; Xin et al. 2011] to scientific visualizations [Bailey 2005]. Second, computer graphics techniques have been employed in developing novel additive manufacturing processes such as deformable objects [Bickel et al. 2010].

One particular affordance of SLS is that it can print joints that do not need assembling. Thus a thread of work from the mechanical engineering community addresses the additive printing of mechanisms. Lipson et al. [2005] print a wide variety of historical mechanisms and note that because additive manufacturing is not as precise as other manufacturing processes, designs must be adapted to the new process, and the mechanism might not function as precisely or as smoothly as desired. Won et al. and Mavroidis et al. report success in the use of additive manufacture of joints that form robots [Won et al. 2000; Mavroidis et al. 2001]. Members of that team then develop a full hand model with joints [Laurentis and Mavroidis 2004]. Their concern is with the successful manufacture of such an object; they are not concerned with automating the process of design, nor in the aesthetics of final object. In our work we will be additionally concerned with the properties of joints such as friction. To date this has not received much attention in the literature; the closest work are techniques to predict the mechanical properties of objects [Rajagopalan and Cutkosky 2003; Chen and Zhezhen 2011].

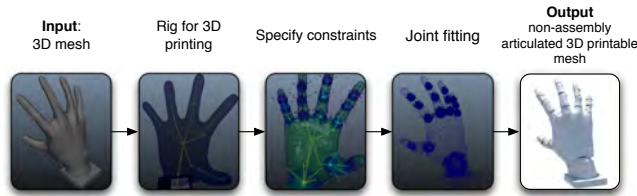
The designs in the previous paragraph are manually created. Our aim is to automate parts of the design process, requiring only rough user annotations on placement and angular freedom of each joint. A seminal work in automated design for additive manufacturing is Lipson & Pollack’s work on automated design of robots with joints and actuators that can locomote [Lipson and Pollack 2000]. This work is a physical analogy of Sims’ well-known work in computer graphics on evolution of jointed virtual creatures [Sims 1994]. However this process evolves the structure as well as joints, whereas we are interested in fitting joints to a given model. Recent advances in shape analysis have provided automated techniques for models with detectable parts [Mitra et al. 2010; Lau et al. 2011], but we address models that are organic in form, and thus do not contain obvious parts. A related work attempts to find articulations of shapes [Xu et al. 2009], but that technique targets deformation of objects, whereas we need to find joints between rigid objects.

Concurrent work [Bächer et al. 2012] explores the same problem of creating a printable articulated model from an input mesh. Their approach takes a skinned mesh as input, i.e., containing data on deformations, which allows for individual rigid parts to be extracted and joint locations to be estimated. This can lead to many joint candidates and therefore collisions need to be resolved. An optimization step is added for consolidating the joint strength. In contrast, our work takes a simple mesh as input and focuses on offering an intuitive interface for users to create an articulated model with a real-time interface, while keeping key concepts about joint design at the forefront and offering added control over the range of movement capable by the joint. Similar to Bacher et al. our universal candidate for all articulations is the ball joint that we custom-shape to enable or disable certain movements. Additionally, we propose a joint design for quick excess material drainage that is suitable for most 3D printing technologies.



Figure 2: A range of test joints fabricated to informally explore physical properties of various joint designs.

3 Overview



The following sections describe our approach for creating 3D-printable figurines. In Section 4, we first derive a generic joint design suitable for articulated figurines. This joint type is then exclusively used to interactively rig a single, manifold input mesh for 3D printing (Section 5). After the user interactively defines rotational constraints, we automatically determine joint attributes, such as size and embedding into the input geometry. The output of this is a 3D printable, posable mesh. Details on the underlying algorithms are provided in Section 6.

4 Generic Joint Template

While a vast selection of mechanical joint designs for kinematic pairs exists, we have a specific set of design criteria for the joints – geared towards our application:

- wide range of angular positions
- absence of lock configurations
- intuitive parameterization to define articulation constraints
- controlled friction to maintain poses under gravity
- assembly-free printing: the joint should leave the 3D printer ready to use
- compact appearance to blend well into the model’s shape

Furthermore, we aim at choosing a single but versatile design, in order to reduce the complexity of the interactive design process. In the remainder we discuss the design decisions that led to our generic template joint used for all joints in our system.

Base design In order to determine the general form of our generic joint, we revisited common joint designs for their suitability (Figure 2).

A popular choice in robotics is the revolute joint with one rotational degree of freedom. By serializing multiple revolute joints, higher degrees of freedom can be achieved. While well suited for actuated, computer-controlled kinematic chains, serialized revolute joints appear less suited for our scenario: the combination of three or more passive revolute joints (required for 3 degrees of freedom) leads to gimbal lock, and the net articulation constraints can only indirectly be designed by controlling the angular range of each in-

dividual axis. We nevertheless evaluated revolute designs by 3D printing them in SLS process.

Traditionally, revolute joints are combined in a gimbal construction; however, the limited resolution of 3D printers would make it hard to constrain gimbal rotation. Instead, we considered a ball-shaped joint from two hemispheres, shown to the right. Rotation between the two hemispheres is controlled by an internal guide-way; the two remaining axes are constrained through the extent of the guiding slits. We believe that our arrangement allows for reasonably intuitive control of rotational constraints, but experiments with a printed sample showed that, in addition to locking, friction varies greatly depending on the joint position, which makes it less suitable for manual articulation.

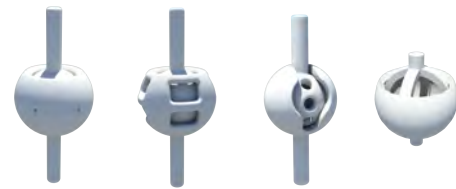


A special case of a revolute chain is the universal, or Cardan, joint (left), which provides two rotational degrees of freedom and blocks in-axis rotation. Aesthetically suitable for figurine modeling, the in-axis constraint makes it unrealistic for many anatomical joints, and it is not obvious how to add friction or custom motion constraints.

The solution we decided to use builds upon a regular ball-and-socket joint (right). It spans a wide range of angles, is free of lock configurations, and allows for intuitive free-form constraints on rotations away from the central axis by shaping the socket opening. The spherical shape blends well with the surrounding model surfaces.



Printability A key challenge to overcome when 3D-printing joints is factoring in the removal of excess material—be it unsintered powder or other support material. Especially in the case of the ball-and-socket joint, powder tends to get stuck in the thin gap between ball and socket, making it impossible to budge without applying a great amount of force, and soluble support materials may be hard to reach by the solvent. To address this issue, we experimented with openings in socket and ball (Figure 3), eventually settling for a cage-in-socket design, Figure 3d. This design has been chosen



(a) Holes in socket (b) Cage in socket (c) Holes in the ball (d) Cage in ball

Figure 3: Powder draining methods we considered.

mainly for aesthetic reasons, as it preserves the smooth appearance of the socket and offers great accessibility to the gap between cage and socket.

Friction Lastly, the printed joints have to exhibit internal friction, just enough to resist gravity to allow for static poses. Usually, for assembled ball joints, this is achieved by pre-stressing the sockets, exploiting elasticity of the materials, which unfortunately cannot be implemented in an assembly-free design. The introduction of friction is further hampered by limited printer resolution: any 3D printing process requires a minimal tolerance between kinematically independent pieces for them not to fuse during the printing process.

Our solution is to extend the cage-in-socket design previously identified as desirable for printability: we slightly increase the diameter of the two bands building the cage, carving corresponding cavities out of the socket to prevent fusing during printing (see Figure 4a). Once the excess material has been removed, the cavities hold the ball in a static positions. As soon as the joint is rotated, the bands leave their cavities and slide onto the socket surface, where now friction holds them in place (Figure 4b). Thus, the joint can be statically posed under any rotation.

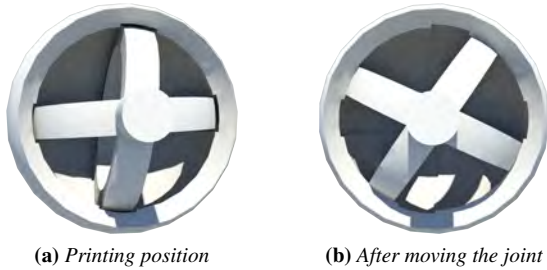
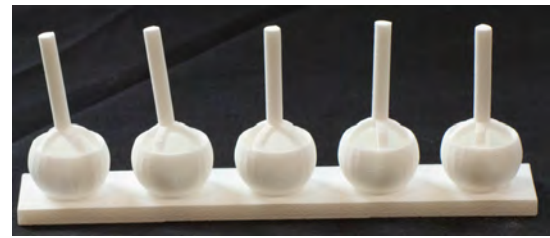


Figure 4: Adding friction to the ball joint. The bands are printed in a position where they are lodged in the two ridges that run along the inside of the socket. When that ball moves it will make direct contact with the socket surface thus providing friction.

Calibration Given this final joint design, tolerances and dimensions have to be chosen to prevent fusion, meet stability criteria, and provide the desired degree of friction. While, in theory, these parameters may be derived from known material parameters and from the specification of the 3D printing process, we found that one and the same process may lead to different results, as individual printers may vary. Thus, we leave determination of joint dimensions to a calibration process that we run on each printer before model generation. For this purpose we created three datasets that we used for calibration by varying three different parameters: tolerance, band thickness and outer radius of the joint, which can be seen in Figure 5. The tolerance in this case is the gap between the bands forming the ball and the groove in the socket as seen in Figure 4. Figure 5a shows a tolerance calibration set where the gap size varies from 0.2 to 0.6 mm. The thickness of the ball bands is crucial for conveying friction, ensuring good powder drainage and keeping structural integrity. A small band size will make powder draining easier but in the same time decrease the friction surface and the solidity of the structure. Conversely, if the thickness increases, excess material removal will become difficult up to the point of not being able to move the joint. The band thickness calibration set contains six samples ranging from 2 to 12 mm in band size (from very thin to virtually covering the sphere) while all other parameters are kept constant (Figure 5b). Finally, the minimum size for a printable and functional ball joint is determined using the size calibration set illustrated in Figure 5c where the radius of the outer ball varies from



(a) Tolerance calibration set



(b) Band thickness calibration set



(c) Size calibration set

Figure 5: Calibration sets for testing 3D printer parameters for achieving optimal friction, excess material clearance and structural integrity preservation. For this purpose we vary three parameters: tolerance, band thickness and joint size.

5 mm to 2 cm and the band thickness is scaled proportionally to this radius.

5 Workflow

The first step in our rigging process for 3D printing is the manual insertion of a classical animation rig into the input mesh, see Figure 6. To facilitate this, we integrated our processing pipeline in Maya, which provides powerful rigging facilities. It would be possible to use preexisting rigs as used for character animation; however, we found that particularly professional animation rigs contain excessive numbers of joints that do not serve a kinematic purpose but control local mesh deformations, such as muscle bulging. In our experience, re-rigging a character from scratch is quicker than “cleaning up” a professional animation rig.

Once the rig is in place, we automatically insert a generic joint template at each joint position of the rig (leaving out “virtual nodes” of the rig where more than two bones are connected, e.g., the center of the hand in Figure 6), scaling each joint to match the surrounding geometry. Where aesthetic reasons suggest other placements, the user has the opportunity to move and scale each joint accordingly.

The central user interaction now consist of specifying the rotational constraints at each joint. We visualize extremal joint rotations as *pins* that denote the directions under which the shaft of the inner cage (and with it, the attached animation rig bone) leaves the socket; see Figure 7. Starting from default pin positions (four pins, 45° off the adjacent bone), the user thus shapes the rotational constraint of each joint. New pins can be inserted, and the extremal shaft directions between pins are interpolated using spherical linear interpolation (Slerp) [Shoemake 1985]. Upon request, the system provides



Figure 6: Hand model rigged for 3D printing.

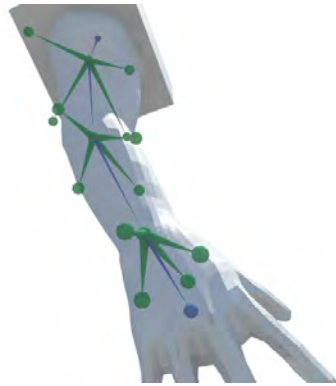


Figure 7: Pins for specifying rotational constraints.

feedback whether the current set of pins is in a physically valid configuration or whether their envelope would cause the socket to widen far enough that the inner cage would fall out, see Figure 8.

Once all pins are set, the system automatically splits the mesh into its components, inserts and shapes the joint and removes excess geometry surrounding the joint to make way for physical movement. If required, the user can undo this step, reposition pins, and re-run this generation of the final mesh.

While the presented workflow is kept as lean as possible, refinements to the process would still be possible. For instance, one could initialize pin positions according to potential constraints defined in a professional animation rig. That said, very few rigs come with constraint information, and as discussed, we found it to be the quicker workflow to re-rig meshes on input. An alternative way to define constraints would be to sequentially pose the model in extremal articulations, inferring the constraint pins from their joint configurations. For longer kinematic chains, however, this mode of constraint specification would require an increasing amount of poses to exhaustively sample the space of joint configurations.

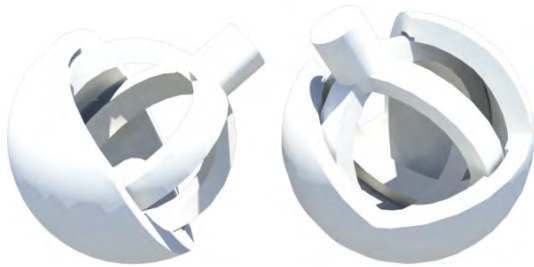


Figure 8: Different configurations of joints showing both invalid (left) and valid (right) joint designs. Invalid designs are flagged to the user.

6 Implementation

Our interactive workflow is supported by a set of algorithms that help placing, scaling and shaping each joint and the adjacent mesh.

Constraint validity The user interface contains a simple option to check the physical validity of the pin constraints. The key test is whether the joint’s inner cage would easily fall out of the socket. To this end, we compute the convex hull of the inward-facing surface of the sculpted socket and check if the center of the spherical inner cage is contained within that hull. If it is, the cage is secured by the socket and thus the configuration is valid.

Joint types Even though we have a single, generic joint template, its insertion differs depending on its context in the model. We distinguish between two cases: *limb joints* that connect two bones that are part of the same, near-cylindrical limb (like a finger joint), and *attachment joints* that connect a bone to a non-cylindrical section and thus appear partially embedded in that body (like in the case of a knuckle joint that connects a finger to the palm). The distinction of these two types is hidden from the user and only used to internally choose the appropriate heuristic when embedding the joint geometry.

We discern the two types by considering the eccentricity of the intersection of the surrounding mesh with the bisecting plane spanned by the two adjacent bones: If the intersection curve is near-circular (if its minimum and maximum distances to the joint center do not differ by more than 40%—a threshold that worked well for our models), we assume a limb joint, otherwise we treat it as an attachment joint.

Joint scale In the case of limb joints, we cast rays from the joint center through the surrounding geometry to obtain point samples of the original mesh that needs to be approximated by the new joint. As the joint is spherical, we fit a sphere (using RANSAC [Fischler and Bolles 1981]) to this point cloud to determine the joint’s radius. For further stability, we limit our ray casting to rays that are at least 45° away from the nearest bone. A result of this sphere fit can be seen in Figure 9.



Figure 9: The result of our sphere fitting on a simple mesh; green points indicate intersection points.

Attachment joints require special consideration, as the direct vicinity of the joint does not allow to meaningfully fit a sphere. Instead, we move the origin of our ray casting along each bone, until one position at the bone fulfills the limb joint criterion; maximally half way along the bone. From here, we obtain point samples and fit a sphere to determine the joint’s radius. If none of the bones contain a point with near-cylindrical environment, we fit a sphere at the joint position nevertheless; if both bones allow for a sphere fit, we choose the one with the smaller radius. In our experiments, this procedure reliably achieved aesthetically pleasing joint sizes for any attachment joints. It is further essential to assign socket and shaft to the two bones so that the shaft points in the direction of the bone used for sphere fitting, thus embedding the socket in the mesh that the limb is attached to.

Lastly, joint scale is partially a question of aesthetic preference. Figure 10 shows three differently scaled wrist joints, at 120%, 100%, and 80% of the sphere fit radius.

Joint shaping Our generic joint template is composed of two disjoint moving parts: the female part (consisting of a hollow ball) and the male part (consisting of a cage ball attached to a shaft). To



Figure 10: Three differently scaled wrist joints, at 120%, 100%, and 80% of the sphere fit radius.

implement the rotation constraints, we dilate the generalized cone spanned by the constraint pins by half a shaft thickness and subtract it from the socket ball using constructive solid geometry (CSG).

Mesh segmentation As each articulation joint links two logically separate mesh parts, the input mesh is segmented accordingly. For limb joints, the geometry is split along the bisecting plane between the adjacent bones; for attachment joints, we split the mesh along the first intersection of the generalized cone with the mesh.

Removing excess geometry Before inserting the joint into the mesh geometry, any parts that would interfere with the joints mechanical function need to be removed. First, we subtract a sphere of the joint’s socket size from the two mesh segments involved. Then, we move one of the mesh segments around the other, rotating it according to the joint parameters, following the trajectory of the extremal positions defined by the pins. While doing so, we perform CSG subtraction to carve out portions of the static mesh segment that otherwise would collide with the moving part (see Figure 11). Lastly, we insert the joint geometry into the model. By choosing a joint shaft that extends beyond the joint’s spherical shape, socket and shaft will be naturally embedded into the adjoining object parts once the model is printed.

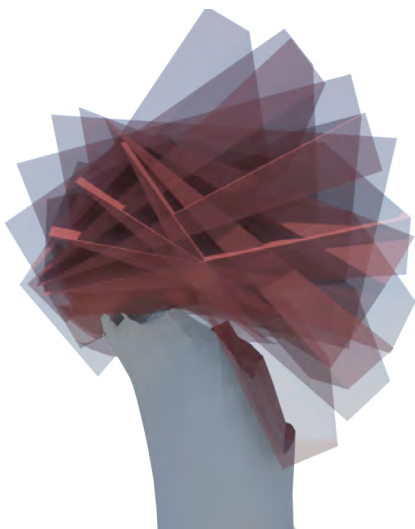


Figure 11: Illustrating excess geometry removal for one object part by rotating an adjoint part (shown in translucent red) into its extremal poses while removing material from the first part (shown in grey).

7 Results and Analysis

In order to validate our results we used two rapid prototyping processes to realize the calibration sets as well as the articulated models: SLS using an EOS FORMIGA P 100 printer, and SLA using an Objet PolyJet™ printer.

Calibration The printed calibration sets (Figure 5) demonstrate the effect of the parameters described in Section 4. The tolerance set (Figure 5a) shows that a gap of 0.2mm is too small as it causes the bands and socket to fuse together. With a 0.3mm gap the joint is easy to manipulate whilst maintaining good friction. Given that the gap is only present between the bands and their underlying grooves, when the ball is moved out of its resting position friction immediately appears. Thus, even for large gap tolerances, such as 0.6mm, the printed model still maintains poses well even when rotated out of its rest position. These observations apply to both rapid prototyping techniques we have employed.

We determined that the minimum size for producing functional joints was 5mm in radius for both printing processes. On this set we have also noticed a difference in material strength and elasticity between the two printers. The samples printed using PolyJet™ are more fragile and brittle at small scales and for very thin features. Even though they exhibited full functionality, applying a greater force caused the thin bands to snap. The material used in SLS printing is stronger and more elastic, thus preventing the bands of the joints from easily breaking.

Band thickness heavily influences the ease of excess material removal and implicitly affects unblocking the joint. Figure 5b illustrates our calibration set for this parameter. As the band size increases, the friction surface also becomes larger requiring much greater force to rotate the joint. For the SLS set we have successfully unblocked the first four joints (increasing band thickness) and in the PolyJet™ set we were only able to move the first two. As all models printed using the latter method require varnishing, this can cause the adherence between adjacent joint parts to increase making them more difficult to unblock. Furthermore, when trying to twist the joint with the thinnest bands they snapped and broke. On the other hand, the SLS powder-based material is more porous and facilitates the joint unblocking process, but it is also more elastic making joints more resistant to twisting.

Articulated Models We have applied our method to models of varying geometrical complexity and number of joints to be articulated, see Figure 12, where we always show the input model, the model fitted with joints, and the final 3D-printed model. The first model is an arm with shoulder, elbow and wrist joints demonstrating a simple kinematic chain; thus this model contained only joints that are considered limb joints. A more complex example is the model of a hand, which apart from the limb joint in the wrist and finger articulations, this model contains attachment joints, showing that our system can handle articulating more than cylindrical geometry. Our final example is an elephant model, which has attachment joints at the base of its legs, with the rest of its articulations being classified as limb joints.

One of the major contributions of this paper is the friction model. Figure 14 shows a photograph of two printed arm models, the left one being printed with friction bands, the right one without, depicting the floppiness of the right model and the ability of the left one to maintain any desired pose. Figure 13 shows the arm model with friction bands in several different poses.

The arm model was printed using both SLS and PolyJet™ at a scale of $47.7 \times 141.9 \times 42.0\text{mm}^3$. Both printed models are fully functional and easy to manipulate while maintaining a balanced level of



(a) Arm model (input, articulated, 3D print)



(b) Hand model (input, articulated, 3D print)



(c) Elephant model (input, articulated, 3D print)

Figure 12: Results of joint fitting on three input models.

friction to allow for posing. However, after several manipulations, one of the joints from the PolyJet™ arm snapped when trying to twist it, which confirms our findings of the calibration set. It may be possible to reduce this problem via stress relief [Stava et al. 2012], but we have not explored this. Depending on the printing process, continued manipulation of a joint can lead to reduced friction due to wear and tear. We have particularly noticed this for SLS, where the powder-based material wears off more easily.

While friction within our joints is roughly isotropic, the lever of the adjustment (length of a limb for bending rotations; thickness of a limb for in-axis twist) may create different degrees of resistance, when manually posing a model. We found this a welcome feature, as it naturally makes it harder to twist the joint than to bend it. Strictly speaking, however, the current joint design does not directly control twist (in-axis rotation). As discussed earlier, the Cardan joint is a possibility to do so, but in the interest of choosing a single joint design, we do not support it in our system.

8 Conclusions

In this paper we have described an approach for semi-automatically fitting printable joints to 3D models. We have described a simple specification to be used for rigging models for 3D printing and detailed our decision process of selecting a maximally versatile joint



Figure 14: Photographs of 3D printed models: left and right models with and without friction joints respectively.

design and steps we have taken to optimize this chosen design for 3D printing. Our approach of fitting printable joints involves mesh segmentation based on joint locations and orientations, while the degrees of freedom of each joint are modeled through the use of computational geometry operations, both in shaping the outer casing of the ball-joints and in sculpting the mesh parts connected by each joint. Finally, our results show that our method can handle different types of input meshes in terms of geometrical complexity and variety and number of articulations.

Acknowledgments

We would like to thank Martin Watmough and Richard Beckett from the Bartlett for their help with printing, and the anonymous reviewers for their valuable feedback. The work was in part funded by the EngD VEIV Centre for Doctoral Training.

References

- ALAVALA, C. R. 2010. *CAD/CAM : Concepts and Applications*. Prentice-Hall.
- AUTODESK, 2012. *Maya User Guide: Rigging*.
- BÄCHER, M., BICKEL, B., JAMES, D. L., AND PFISTER, H. 2012. Fabricating articulated characters from skinned meshes. *ACM Trans. Graph. (Proc. SIGGRAPH)* 31, 4 (July), 47:1–47:9.
- BAILEY, M. 2005. Layered manufacturing for scientific visualization. *Commun. ACM* 48, 6 (June), 42–48.
- 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, 4 (July), 63:1–63:10.

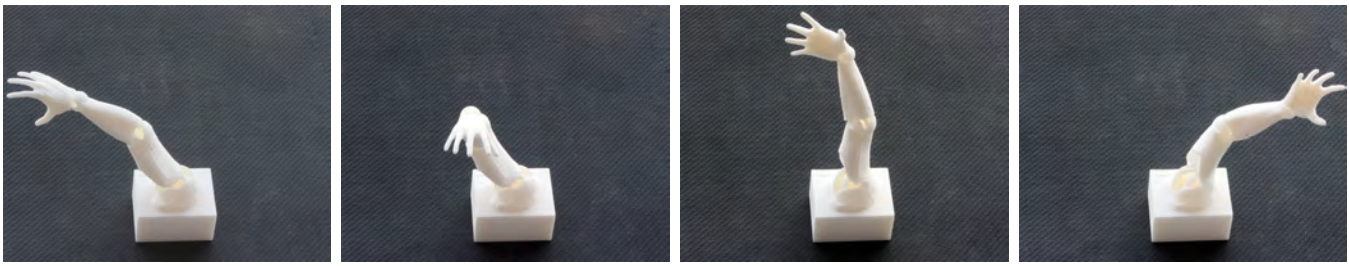


Figure 13: Photographs of 3D printed fitted arm model articulated in different poses.

- CHEN, Y., AND ZHEZHENG, C. 2011. Joint analysis in rapid fabrication of non-assembly mechanisms. *Rapid Prototyping Journal* 17, 6, 408–417.
- DIMITROV, D., SCHREVE, K., , AND DE BEER, N. 2006. Advances in three dimensional printing state of the art and future perspectives. *Rapid Prototyping Journal* 12, 136–147.
- FIGUREPRINTS LLC, 2012. Figureprints. <http://www.figureprints.com>.
- FISCHLER, M., AND BOLLES, R. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24, 6, 381–395.
- GIBSON, I., ROSEN, D. W., AND STUCKE, B. 2009. *Additive Manufacturing Technologies: Rapid Prototyping to Direct Digital Manufacturing*. Springer.
- KODAMA STUDIOS, 2012. My robot nation. <http://www.myrobotnation.com/>.
- 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.
- LAURENTIS, K. J. D., AND MAVROIDIS, C. 2004. Rapid fabrication of a non-assembly robotic hand with embedded components. *Assembly Automation*, 24, 394–405.
- LEE, K. 1999. *Principles of CAD/CAM/CAE Systems*. Prentice Hall.
- LEONDES, C. T., Ed. 2003. *Computer Aided and Integrated Manufacturing Systems, Volume 4: Computer Aided Design / Computer Aided Manufacturing (CAD/CAM)*. World Scientific.
- LIPSON, H., AND POLLACK, J. B. 2000. Automatic design and manufacture of robotic lifeforms. *Nature* 406, 6799, 974–978.
- LIPSON, H., MOON, F., HAI, J., AND PAVENTI, C. 2005. 3-d printing the history of mechanisms. *Journal of Mechanical Design* 127, 123–132.
- LO, K.-Y., FU, C.-W., AND LI, H. 2009. 3D polyomino puzzle. *ACM Trans. Graph.* 28, 5 (Dec.), 157:1–157:8.
- MAKIELAB, 2012. MakiLab. <http://www.makielab.com/>.
- MAVROIDIS, C., DELAURENTIS, K., WON, J., AND ALAM, M. 2001. Fabrication of non-assembly mechanisms and robotic systems using rapid prototyping. *ASME Journal of Mechanical Design* 123, 4, 516–524.
- MCMAINS, S., BAILEY, M., AND CRAWFORD, R. 2005. Layered manufacturing as a graphics display device. In *ACM SIGGRAPH 2005 Courses*, ACM, New York, NY, USA, SIGGRAPH '05.
- MITRA, N. J., AND PAULY, M. 2009. Shadow art. *ACM Trans. Graph.* 28, 5 (Dec.), 156:1–156:7.
- MITRA, N. J., YANG, Y.-L., YAN, D.-M., LI, W., AND AGRAWALA, M. 2010. Illustrating how mechanical assemblies work. *ACM Trans. Graph.* 29 (July), 58:1–58:12.
- RAJAGOPALAN, S., AND CUTKOSKY, M. 2003. Error analysis for the in-situ fabrication of mechanisms. *Journal of Mechanical Design* 125, 4, 809–822.
- SHOEMAKE, K. 1985. Animating rotation with quaternion curves. *ACM SIGGRAPH computer graphics* 19, 3, 245–254.
- SIMS, K. 1994. Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '94, 15–22.
- STAVA, O., VANEK, J., BENES, B., CARR, N., AND MĚCH, R. 2012. Stress relief: improving structural strength of 3D printable objects. *ACM Trans. Graph. (Proc. SIGGRAPH)* 31, 4 (July), 48:1–48:11.
- STEMP-MORLOCK, G. 2010. Personal fabrication. *Commun. ACM* 53, 10 (Oct.), 14–15.
- THINGIVERSE, 2012. <http://www.thingiverse.com/search?q=gears>.
- THOMPSON, R. 2007. *Manufacturing Processes for Design Professionals*. Thames & Hudson.
- WON, J., LAURENTIS, K. J. D., AND MAVROIDIS, C. 2000. Rapid prototyping of robotic systems. In *ICRA*, IEEE, 3077–3082.
- XIN, S., LAI, C.-F., FU, C.-W., WONG, T.-T., HE, Y., AND COHEN-OR, D. 2011. Making burr puzzles from 3D models. *ACM Transactions on Graphics (SIGGRAPH 2011 issue)* 30, 4 (August), 97:1–97:8.
- XU, W., WANG, J., YIN, K., ZHOU, K., VAN DE PANNE, M., CHEN, F., AND GUO, B. 2009. Joint-aware manipulation of deformable models. *ACM Trans. Graph.* 28, 3 (July), 35:1–35:9.