

Fabricating and Rigging Creative Characters

Abstract

Creative modeling provides a new means for novice users to create 3D content in an open-end manner. However, most current creative modeling methods are mainly designed to model static objects only. In contrast to these methods, we present a new method for modeling dynamic creative models which are rigged and fabricatable. Starting from a small set of skinned watertight objects, our system iteratively synthesizes new creative characters for users to explore. A user can choose those of interest for animation or fabrication directly. By unifying modeling, animation and fabrication together, our method provides an efficient and convenient means for users to facilitate the creative design process.

Keywords: Creative modeling, Surface merging, Fabrication, Skinning

1. Introduction

Designing 3D models efficiently and creatively is always a challenging problem in the computer graphics community. Sketch-based methods [1, 2] provide a rather convenient manner, but are mostly limited to the creation of smooth rotund objects. Recent trends emphasize more and more on the creativity aspect of the modeling process. Geometric modeling is treated as not only a craft by also an open-ended creative process. Various techniques were presented to support creative discovery in 3D modeling [3, 4, 5, 6, 7, 8].

However, the creation of a 3D model is only the beginning. In most applications, we need the model to be animated, and thus it must be rigged. Traditionally, the two steps (modeling and rigging) are separated, which makes the incremental update of the shape especially difficult. Animators need to frequently switch between modeling and rigging tools. In [9], Borosan et al. presented a unified approach by integrating modeling and rigging together. Jin et al. [10] further incorporated motion retargeting into the unified process, to provide a non-linear modeling/animation tool. It is especially important to combine creativity-support modeling and animating into a unified framework. Therefore, artists can immediately animate the model in an incremental manner, and observe the rationality of the animation. It can also save unnecessary repeated work by reusing existed skinned models. Such a method will find its important applications in computer animation, computer game and digital entertainment. However, few, if any, research efforts on this direction have been reported to date, to the best of our knowledge.

33 Besides, the popularity of 3D printing techniques
34 makes the fabrication of custom-made 3D models at
35 home possible. Incorporating fabrication analysis in a
36 creative modeling and animating framework will signifi-
37 cantly benefit designers. They can not only construct
38 stunning models by combining different parts together,
39 but also explore the physical feasibility of various com-
40 binations under different poses. However, many factors
41 need to be taken into consideration before a 3D mod-
42 el can be printed out, such as topology, balance, struc-
43 ture and so on. For our case, the challenges are even
44 more serious. Firstly, the skeleton of a skinned mod-
45 el usually contains many joints to enhance expressive-
46 ness. However, having many joints becomes a burden
47 for fabrication and is usually unnecessary. For example,
48 in Figure 11(c), we actually do not need so many joints
49 in the tail when fabricating. Another important issue is
50 that the fabricated model should maintain balance under
51 a sequence of poses. Although researchers have previ-
52 ously investigated the fabrication of articulated charac-
53 ters [11, 12] and the balance of fabricated shapes [13],
54 the balancing issue of fabricated 3D models under mul-
55 tiple poses has not been addressed yet.

56 In this paper, we present the first system for creative
57 modeling of fabricatable skinned objects. As illustrated
58 in Figure 11, starting from a small set of skinned water-
59 tight models, our system evolves from the current set
60 into a new group of diverse shapes by applying various
61 structure alteration tools [8]. Users can then explore the
62 suggested shapes to select those of interest to form a
63 new evolving set. The evolution process repeats until
64 the maximum generation is reached or the user is satis-
65 fied with the result. The user can then choose those of

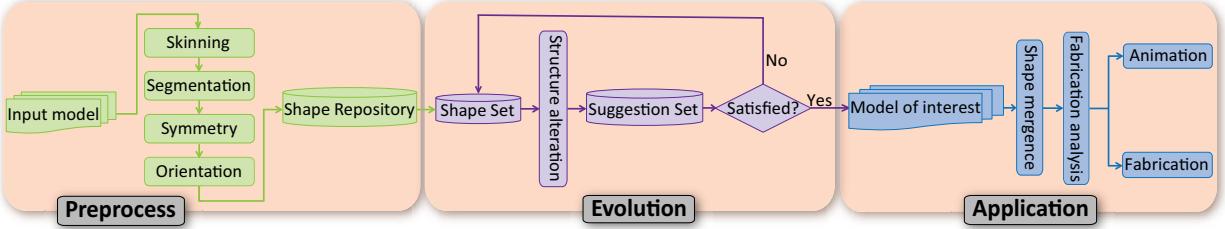


Figure 1: Illustration of our approach.

interest for animation or fabrication. With the seamless integration of modeling, animation and fabrication, our system provides animation-ready and fabrication-ready objects based on the users' design, and thus facilitates the whole creation process. There are two main technical contributions in our system:

- **Mergence of rigged parts.** We present novel techniques to merge both the surface and the skeleton of two rigged parts together, to quickly update the skinning weights in a local manner.
- **Fabrication analysis of rigged shapes.** We propose a fabrication-aware optimization algorithm of skeletons to determine the optimal number of joints and their optimal positions. In addition, we present a novel method to guarantee the balance of a fabricated shape under different poses.

2. Related Work

Part-based shape synthesis and exploration. Due to its efficiency and convenience, part-based shape composition methods receive a lot of attention [14, 15]. As the graphics community gradually realize the importance of providing creativity support in modeling process, researchers began to design various schemes to allow artists explore various possibilities and alternatives during the modeling process [3, 4, 5, 6, 7]. Different from these approaches, we further integrate animation and fabrication in a creative modeling framework. Users can get more inspirations from the animated shapes and can easily validate the feasibility for fabrication.

Rigging. A character in animation is usually represented as a hierarchical set of interconnected bones, and rigging is used to animate articulated objects [16] to bring them to life. Various techniques have been proposed [17] to rig 3D characters. Recently, Borosan et al. [9] introduced a unified framework by combining sketch-based modeling and rigging together. This

work is further extended by Jin et al. [10] to support interleaved animation, modeling, and editing for non-experts. Different from these previous methods, we combine creativity-support modeling, rigging and fabrication together, to facilitate the creative design process.

Fabrication-oriented design. As 3D printers become more accessible in daily life, fabrication techniques and their applications become more and more attractive. However, a lot of work need to be done before a 3D model can be printed out. The model must meet a set of topological requirements [18], be well balanced [13], be structurally strong [19]. All these demands stimulate a lot of fabrication oriented geometric processing research efforts. We investigate the balancing problem of articulated characters under multiple poses, which has not yet been addressed in previous works [11, 12, 13].

3. Approach Overview

The pipeline of our system consists of three stages (Figure 1): Preprocessing stage, evolution stage, and application stage. In the preprocessing stage, a watertight input model needs to go through skinning, segmentation, symmetry detection and co-orientation before being stored in a shape repository. The user then picks up several models as the initial set for the evolution stage. In the evolution stage, the current set of shapes is evolved into a new set by applying various structure alteration tools. If the user is not satisfied with the result, he/she can choose to continue the evolution with the new set as input. Otherwise, the user can stop the evolution step and choose the shapes of interest and move to the application stage. In the application stage, the chosen model is converted to a watertight shape. After the skinning weights are updated accordingly, the model undergoes a fabrication-oriented analysis. Finally, the shape is ready to be used for animation and fabrication.

137 4. Shape Merging

138 Merging is a key process in our framework. Besides
 139 the mergence of surfaces, we also need to merge the
 140 skeletons of two merging parts and update the skinning
 141 weights.

142 4.1. Surface Merging

143 Given two parts placed relative to each other in 3D,
 144 we aim to seamlessly merge the surfaces of them into a
 145 new watertight one. We assume there is a gap between
 146 the given two parts (Figure 2(a)). The gap could be
 147 obtained by shifting one part slightly along the normal
 148 of the fitting plane of the boundary vertices. Such an
 149 assumption provides more flexibilities in the modeling
 150 process: It can deal with the case when two merging
 151 boundaries are not homeomorphic. We first fill it with
 152 a triangle stripe (Figure 2(b)), and then conduct a local
 153 Laplacian smoothing process to achieve a smooth geo-
 154 metric transition and improve the mesh quality (Figure
 155 2(c)).

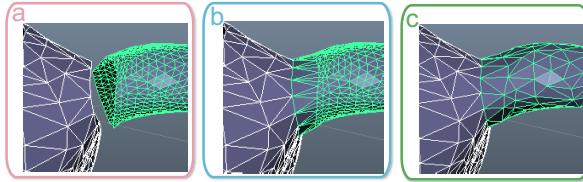


Figure 2: Surface merging. (a) Two input merging parts. (b) Gap triangulation. (c) Boundary smoothing.

156 We formulate such a gap filling problem as an opti-
 157 mal boundary bridge triangulation (BBT) problem [20].
 158 Given two piecewise linear curve C_p (with m vertices)
 159 and C_q (with n vertices), a BBT is an ordered collec-
 160 tion of triangles $M = \{T_1, T_2, \dots, T_N\}$, which is formed
 161 by iteratively applying the P-succeed operator and Q-
 162 succeed operator. The P-succeed operator is used to
 163 form a new triangle $p_i p_{i+1} q_j$ from the edge $p_i q_j$ (Fig-
 164 ure 3(a)). The Q-succeed operator is used to form a
 165 new triangle $p_i q_j q_{j+1}$ from the edge $p_i q_j$ (Figure 3(b)).
 166 Our purpose here is to find one particular BBT from
 167 those BBTs to optimize a given objective function. Ac-
 168 cording to [20], the combinatorial search problem can
 169 be converted into a shortest path problem (Figure 3(c)),
 170 which can be solved by the Dijkstra's algorithm. How-
 171 ever, we still need to develop an appropriate measure for
 172 the graph link. The key consideration of shape merg-
 173 ing is to ensure a smooth transition around the merging
 174 boundary as well as the constructed triangles to be as

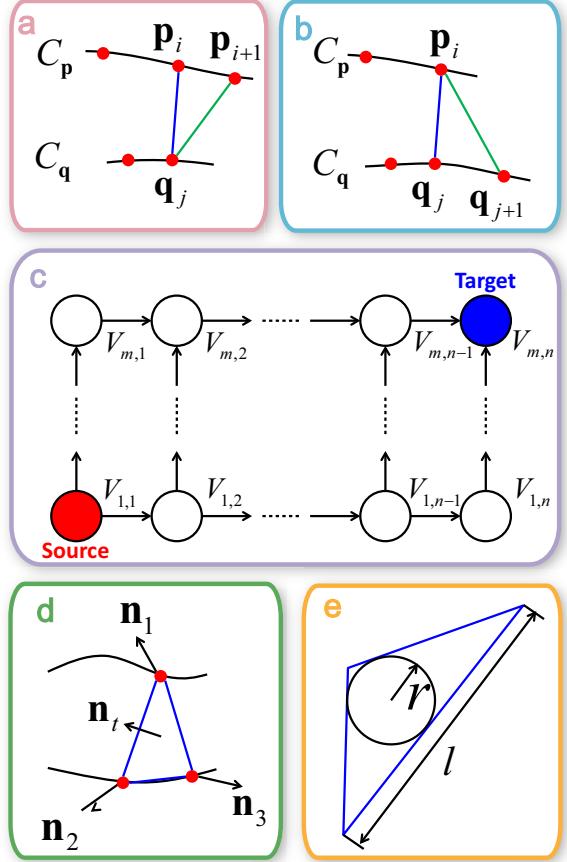


Figure 3: Illustration of the boundary bridge triangulation (BBT). (a) The P-succeed operator. (b) The Q-succeed operator. (c) The shortest path problem for BBT. (d) Illustration of E_s . (e) Illustration of E_r .

175 regular as possible. Therefore, we define the graph link
 176 measure for a triangle T as follows:

$$177 E_m = \omega E_s + E_r,$$

178 where E_s is used to evaluate the consistency of T 's nor-
 179 mal to the averaged normals around T 's merging bound-
 180 ary, and E_r is employed to measure the quality of T .
 181 Here we set $\omega = 0.2$. E_s is defined as follows:

$$182 E_s = 3 - \mathbf{n}_t \cdot \mathbf{n}_1 - \mathbf{n}_t \cdot \mathbf{n}_2 - \mathbf{n}_t \cdot \mathbf{n}_3,$$

183 where \mathbf{n}_t is the normal of T , \mathbf{n}_1 , \mathbf{n}_2 , and \mathbf{n}_3 are the nor-
 184 mals of the adjacent triangles to T , respectively. E_r is
 185 defined as follows:

$$186 E_r = |2\sqrt{3} \times r/l - 1.0|,$$

187 where r is the radius of the inscribed circle, and l is the
 188 maximum length of the three edges of T .

189 **4.2. Skeleton Merging**

190 To merge the skeletons, we first compute the position
 191 on the main shape skeleton that is the closest to the end
 192 joint of the part shape skeleton. If the closet position is
 193 the position of a joint of the main shape skeleton, we di-
 194 rectly connect this joint to the end joint of the part shape
 195 skeleton (Figure 4(a)). Otherwise, we first insert a new
 196 joint in the closest position and then connect the insert-
 197 ed joint with the end joint of the part shape skeleton
 198 (Figure 4(b)).

199 **4.3. Skinning Weights Update**

200 Once the surface and skeleton of two input parts are
 201 merged, we need to compute the skinning weights of
 202 the merged shape. We take a similar local updating
 203 strategy as [9] to first determine those dirty vertices
 204 and bones that need to be updated, and then run a lo-
 205 cal skinning process to generate new skinning weights.
 206 A vertex is considered to be dirty if its closest visible
 207 bone in the composited shape does not have an above-
 208 average weight (taking the average over all positive
 209 bone weights associated with the vertex), and the bone
 210 is considered to be dirty too. We conduct a flooding al-
 211 gorithm starting from the vertices around the merging
 212 boundary and stop when no dirty vertices is found. We
 213 expand 2-rings from the outer dirty vertices to ensure
 214 skinning weights vary smoothly between recomputed
 215 and remaining vertices (see Figure 5).

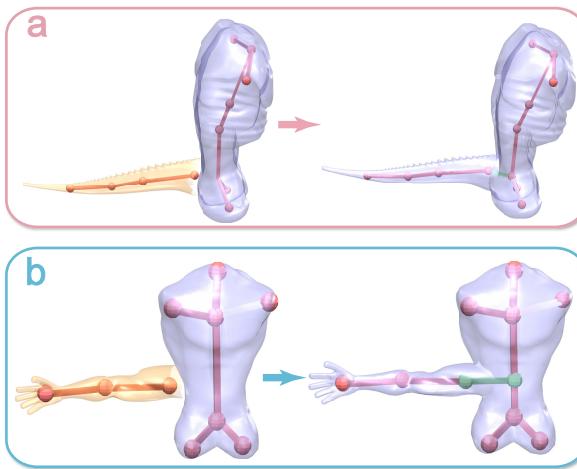


Figure 4: Skeleton merging. The part shape has a light brown color. The inserted joint and the connecting bone is highlighted in green. The end joint of the part shape skeleton is directly connected to the closest joint of the main shape skeleton (a). A new joint is inserted into the main shape skeleton. Then the part shape skeleton is connected to the inserted joint (b).

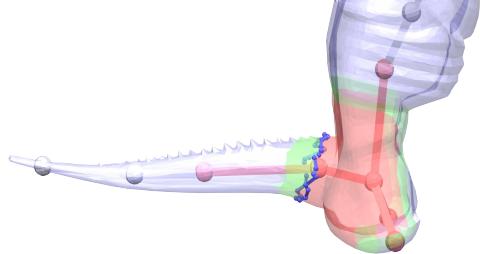


Figure 5: Illustration of dirty vertices and bones. The dirty vertices and dirty bones are marked in red. The expanded vertices are marked in green. The merging boundary is marked in purple.

216 **5. Fabrication-oriented Analysis**

217 In this section, we introduce a fabrication-oriented
 218 analysis process to generate fabrication-ready articulat-
 219 ed models.

220 **5.1. Fabrication-oriented Skeleton Optimization**

221 As mentioned before, the skeleton requirement for
 222 animation and that for fabrication are different. We need
 223 to optimize the joint number and the joints' positions of
 224 the rigged skeleton before fabrication.

225 We follow four principles to decide whether a joint in
 226 the skeleton need to be preserved for fabrication:

- 227 1. The *critical joint* located at the boundary of two
 228 semantic parts should be preserved, because this
 229 joint is considered to be closest to the center of the
 230 part boundary.
- 231 2. If some of the joints are seldom transformed in
 232 most animation sequences, they can be discarded
 233 in fabrication.
- 234 3. The distance between two neighbour joints for fab-
 235 rication should be larger than a threshold.
- 236 4. The joint specified by the user should be preserved.

237 Given a sequence of poses of the model, we define a
 238 variable ρ_i for each joint J_i to evaluate its necessity for
 239 fabrication:

$$240 \quad \rho_i = \sum_k \|\mathbf{M}_k^i - \mathbf{I}\|_F^2,$$

241 where \mathbf{M}_k^i is the k th transformation matrix of J_i , \mathbf{I} is the
 242 identity matrix, $\|\cdot\|_F^2$ is the Frobenius norm. By sum-
 243 ming the differences between the transformation matrix
 244 and the identity matrix in each pose of joint J_i , ρ_i mea-
 245 sures the extent of animation of J_i in the sequence.

246 To determine the optimal fabrication position of a
 247 joint J_i , we collect those vertices V_m whose skinning
 248 weights related to the joint are above a given threshold
 249 ω ($\omega = 0.8$ in our experiments). We then fit a sphere to

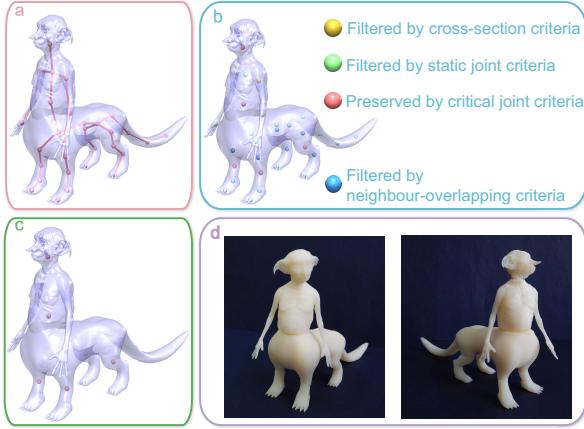


Figure 6: Illustration of our fabrication-oriented joint filtering process. Given a rigged shape (a), our skeleton optimization algorithm trims the skeleton joints (b), resulting in a set of fabrication-oriented joints (c). The Fabricated shape with these mechanical joints are shown in (d).

²⁵⁰ V_m weighted by their skinning weights. The center of
²⁵¹ the sphere is then considered as the optimal fabrication
²⁵² position of J_i , and the radius is the estimated size of the
²⁵³ fabricated joint.

²⁵⁴ We finally decide to remove or preserve the joint J_i
²⁵⁵ according to the following four criteria (Figure 6): (1)
²⁵⁶ the distance d_{ij} between joint J_i and its neighbor J_j is
²⁵⁷ within a given threshold ($d_{ij} < 1.2(r_i + r_j)$, r_i and r_j are
²⁵⁸ the estimated joint radii of J_i and J_j) to prevent physical
²⁵⁹ joints overlapping and J_i is the one with a smaller ρ ; (2)
²⁶⁰ ρ_i is below a given threshold as the joint can be consid-
²⁶¹ ered as static; (3) the joint whose cross section area is
²⁶² smaller than the threshold A_{min} (we use $A_{min} = 0.179$
²⁶³ in our experiments), because the local stress of such a
²⁶⁴ joint is too high, and thus the structure could break; and
²⁶⁵ (4) the joints should be symmetrically distributed on the
²⁶⁶ fabricated object, because our structure alteration tools
²⁶⁷ preserve symmetry while evolving the shape set.

268 5.2. Multiple-pose Shape Balancing

²⁶⁹ We extend the method of [13] to handle the multiple-
²⁷⁰ pose shape balancing problem. There are two main d-
²⁷¹ifferences between our method and theirs: (a) we carve
²⁷² the inner of all posed shapes consistently and the carv-
²⁷³ ing makes all the posed shapes in balance; and (b) we
²⁷⁴ use a hexahedral mesh, adapting to the shape, to repre-
²⁷⁵ sent the volume of each posed shape other than a simple
²⁷⁶ voxel grid.

²⁷⁷ For a given model M , let S_0, S_1, \dots, S_k be the surface
²⁷⁸ of k different poses of the model, our purpose is to solve
²⁷⁹ the following optimization problem by carving the in-

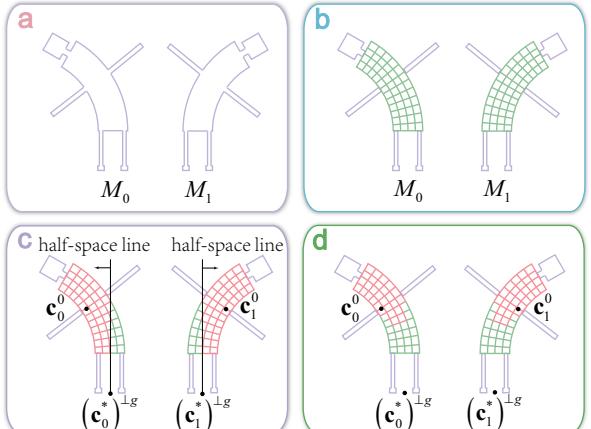


Figure 7: Illustration of multi-pose shape balancing. A 3D model is depicted with a 2D shape (the purple shape in (a)). Given two models M_0 and M_1 (a), we first construct consistent hexahedral meshes for the torso part (depicted by 2D grid) (b). The support points for M_0 and M_1 are identified by $(\mathbf{c}_0^*)^{\perp g}$ and $(\mathbf{c}_1^*)^{\perp g}$ respectively (c). The current center of mass for M_0 and M_1 are denoted by \mathbf{c}_0^0 and \mathbf{c}_1^0 respectively (c). The half-space planes are depicted with the half-space lines (c). The carving voxels for each model are marked with red color. Finally, we sort and cut the intersection (marked in red in (d)) between the two sets of carving voxels.

²⁸⁰ terior volume of the main part (torso part) of the shape
²⁸¹ simultaneously among all the posed shapes:

$$282 \quad \arg \min \sum_k \|(\mathbf{c}(S_k, I_k) - \mathbf{c}_k^*)^{\perp g}\|^2,$$

²⁸³ where I_k is the inner surface corresponding to S_k , $\mathbf{c}(S, I)$
²⁸⁴ computes the center of mass of a model given the outer
²⁸⁵ surface S and the inner surface I , \mathbf{c}_k^* is the target cen-
²⁸⁶ ter of mass of a model in the k th pose, \mathbf{g} is the gravity
²⁸⁷ direction and, ${}^{\perp g}$ denotes the perpendicular projection
²⁸⁸ onto the support plane along the gravity vector.

²⁸⁹ We first map the volume of all the pose surfaces onto
²⁹⁰ a common polycube domain, and construct consistent
²⁹¹ hexahedral meshes for all of them [21]. We then use a
²⁹² heuristic to approximate the optimal voxel carving (Fig-
²⁹³ ure 7). For the k th pose of a model M , let the current
²⁹⁴ center of mass be \mathbf{c}_k^0 , we can calculate a cutting plane
²⁹⁵ through $(\mathbf{c}_k^*)^{\perp g}$ which is perpendicular to $(\mathbf{c}_k^0 - \mathbf{c}_k^*)^{\perp g}$.
²⁹⁶ Carving voxels (denote the set as V_k) in the half-space
²⁹⁷ plane that contains \mathbf{c}_k^0 will potentially bring \mathbf{c}_k^0 closer to
²⁹⁸ \mathbf{c}_k^* and thus may decrease the optimization energy. We
²⁹⁹ collect all the voxels that belong to $\bigcap_k V_k$. We sort and
³⁰⁰ cut voxels in a decreasing order based on their *signed*
³⁰¹ distances to the cutting planes as follows:

$$302 \quad di = \sum_k (\mathbf{v}_i - \mathbf{c}_k^*) \cdot (\mathbf{c}_k^0 - \mathbf{c}_k^*)^{\perp g},$$

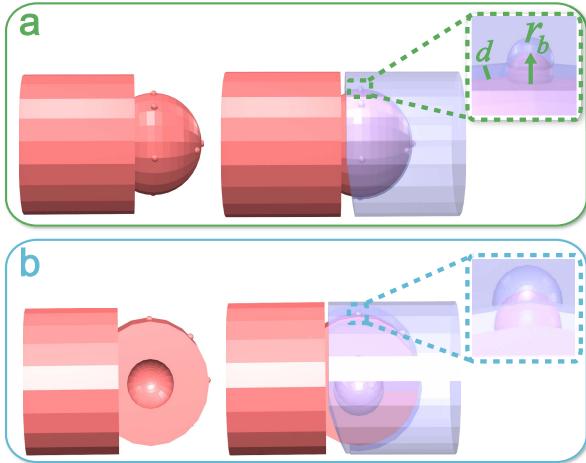


Figure 8: Two types of joints. (a) The ball-and-socket joint. (b) The hinge joint.

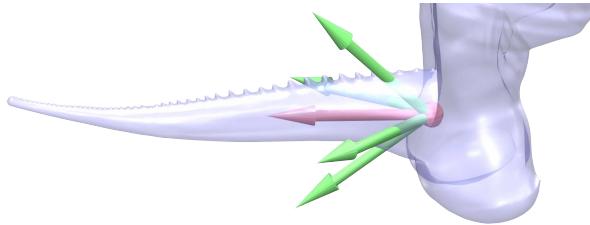


Figure 9: Illustration of pins. The pin in red identifies the direction of the bone. The ball in red denotes the position of the mechanical joint. The pins in green specify the rotation constraints of the mechanical joint. The user manually manipulates the green pins to specify the rotation constraints.

303 where \mathbf{v}_i is the centroid of voxel i .

304 If there is no feasible carving solution over all poses,
305 the user is asked to remove several poses (selected by
306 the user) and the optimization process is repeated.

307 5.3. Printable Joint Design

308 *Joint Type Estimation.* We use two types of mechani-
309 cal joints (see Figure 8): the ball-and-socket joint and
310 the hinge joint. We allow the user to manually identify
311 the rotation constraints at each joint. The user manipu-
312 lates the pins (Figure 9) that denote the directions under
313 which the part shape leaves the main shape. Once the
314 pins are set, the joint type can be automatically calcu-
315 lated.

316 *Posable Joint Construction.* Different from the printing
317 of a static model, a key challenge for articulated mod-
318 el printing is that the model should be able to hold a
319 pose. Joints need to exhibit internal friction to with-
320 stand the gravity. To achieve that, we fabricate small

321 bump spheres of radius r_b (we set $r_d = 0.06$ empiri-
322 cally) onto the positive joint parts and subtract spheres
323 with radius $r_b + d$ (we set $d = 0.05$ empirically) from
324 the negative joint part to prevent the fusion of movable
325 parts during manufacturing as done in [12] (Figure 8).

326 6. Shape Suggestion

327 Similar to [8], we iteratively provide a set of whole
328 shapes for the user to explore in the evolution stage. In
329 each iteration, we evolve the current set of shapes into a
330 new one by applying the structure alteration tools. The
331 structure alteration tools are inspired from the opera-
332 tions artists commonly used to modify shape structures.
333 It includes: (a) Replace tool (Figure 10(a)): Exchanging
334 parts between different shapes; (b) Increase tool (Figure
335 10(b)): Duplicating or triplicating a part; (c) Combine
336 tool (Figure 10(c)): Concatenating the upper body of the
337 humanoid shape to the lower body of the quadrupeds or
338 bird-like shapes to generate a Centaur-like creature; and
339 (d) Insert tool (Figure 10(d)): Adding a new type of part
340 to the current shape.

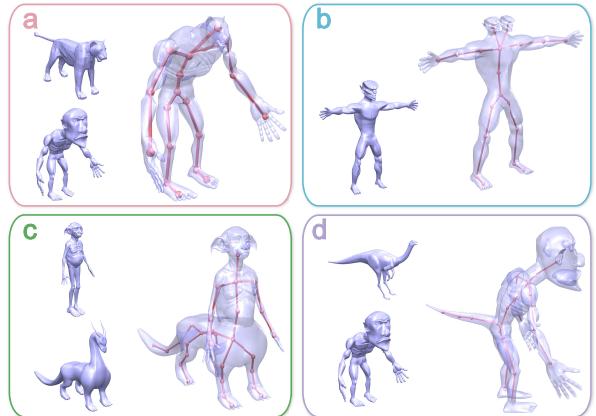


Figure 10: Structure alteration tools. In each sub-figure, we show the original shape(s) (on the left), the generated shape and the merged skeleton (on the right). Illustration of the replace tool (a), increase tool (b), combine tool (c) and insert tool (d).

341 7. Results

342 We built a prototype system on a desktop computer
343 with Intel Core i7 (3.77GHz), 8 GB RAM and NVIDIA
344 Geforce GTX 660 GPU. The system relies on a reposi-
345 tory of watertight skinned models. We currently built a
346 repository by automatically rigging and animating a set
347 of models with the method of [22]. These models are

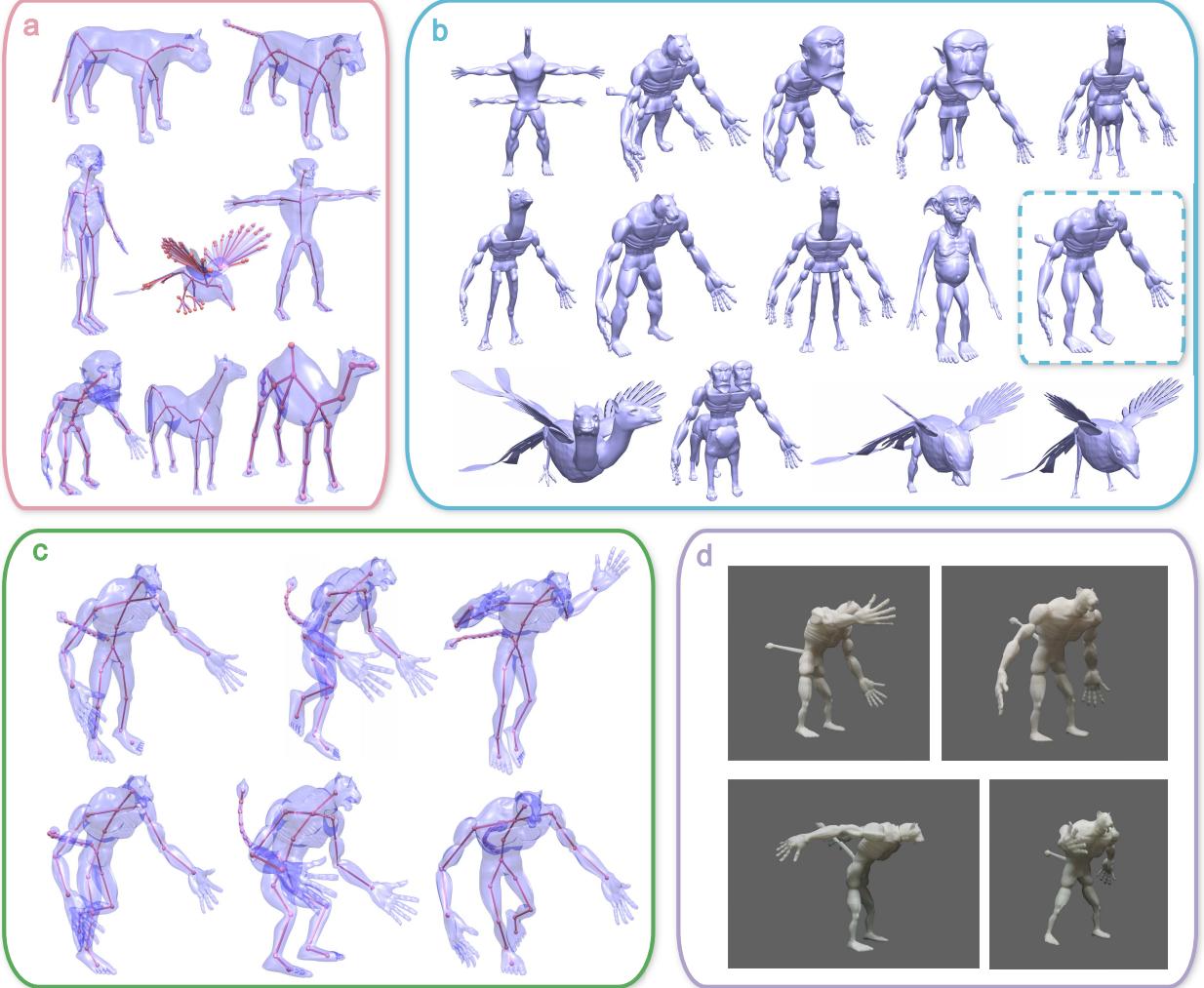


Figure 11: An example of the creative design process. (a) Input models. (b) Some of the shape suggestions provided by our system during the evolution stage. The model enclosed by the dashed rectangle is the model of interest identified by the user. (c) Several poses generated during the animation stage. (d) Three poses of the fabricated model generated during the fabrication stage.

348 then segmented and labeled with consistent skeleton-
 349 s. The segmentation and labeling is automatically ob-
 350 tained [23]. They are also normalized, co-oriented and
 351 symmetry-detected. The whole preprocess took about
 352 25 minutes on average for each shape. Our system can
 353 guarantee the balance of shape under different poses,
 354 see examples in Figure 11(d), Figure 12(d), and Fig-
 355 ure 13(d).

356 We conducted an informal user study by inviting 8 s-
 357 tudents. We first introduced our interface to them. Then
 358 they were given 30 minutes to familiarize themselves
 359 with our interface. Finally we allowed them to use our
 360 system without any limitation on the model type and
 361 time. An example is shown in Figure 12.

362 We recruited 3 artists with 4+ years experience on
 363 3D modeling and character rigging to evaluate our tech-
 364 nique. An example is given in Figure 13. The experts
 365 reacted positively to our system. They agreed that our
 366 system combines modeling, animation and fabrication
 367 into a single coherent framework, making the whole cre-
 368 ation process much easier.

369 7.1. Limitations

370 Either the skeleton optimization or the multi-pose
 371 shape balancing in the fabrication-oriented analysis re-
 372 lies on an assumption that the composed character has a
 373 sequence of poses. If not, the second criteria in filtering
 374 of joints will be simply removed or replaced by manu-

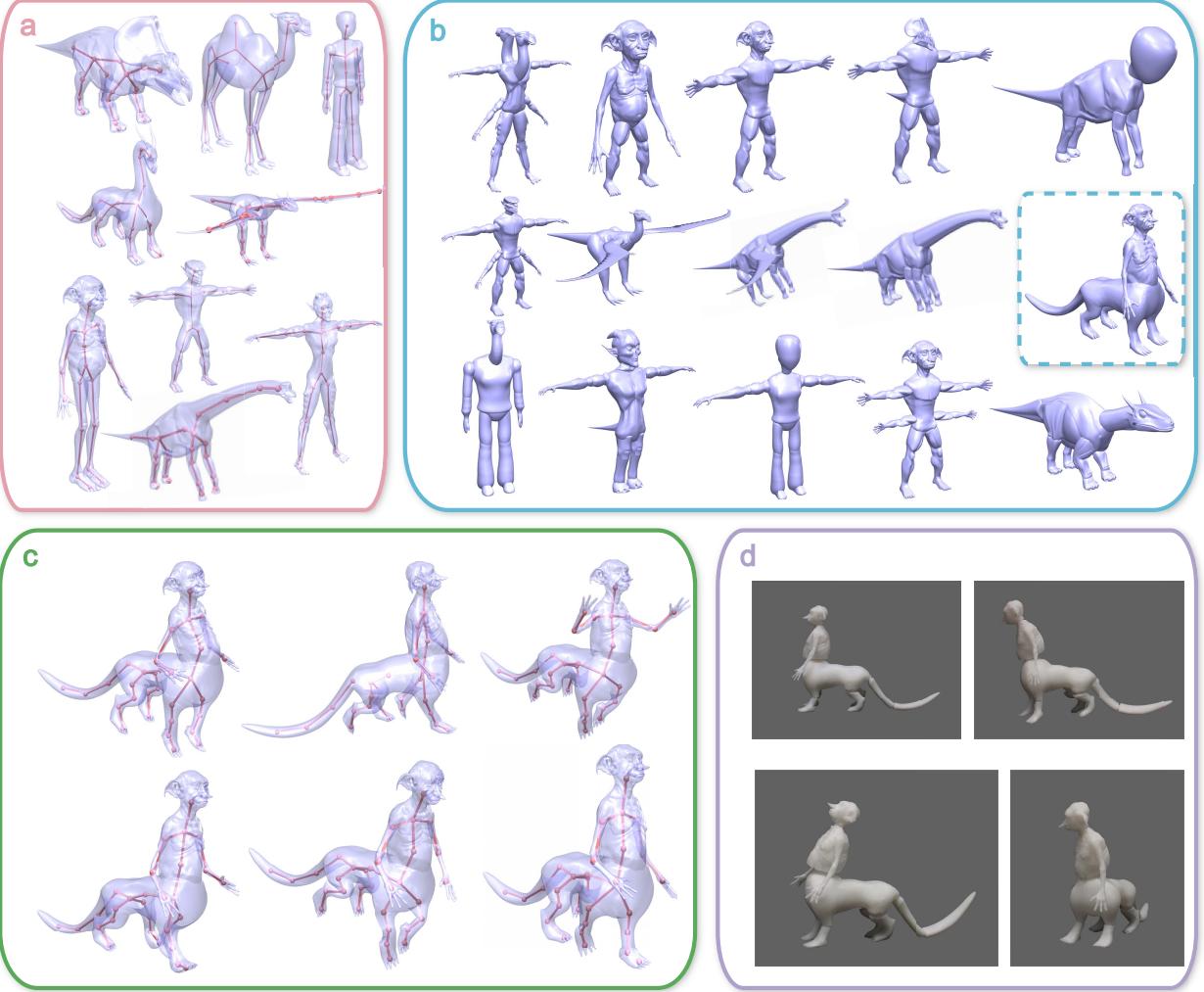


Figure 12: An example of the creative design process. (a) Input models. (b) Some of the shape suggestions provided by our system during the evolution stage. The model enclosed by the dashed rectangle is the model of interest identified by the user. (c) Several poses generated during the animation stage. (d) Three poses of the fabricated model generated during the fabrication stage.

375 al selection. For multi-pose balancing, user can manual
376 select some expected poses or specify the limit of each
377 joint.

378 8. Conclusions

379 In this paper, we present a system to provide cre-
380 ativity support for the modeling of fabricatable skinned
381 objects. With a small set of preprocessed watertight
382 skinned objects, our system iteratively evolves them in-
383 to a new shape set through various structure altera-
384 tion tools. Users can then explore various possible modeling
385 results from the shapes suggested by our system. The
386 process continues until the user is satisfied with the re-

387 sult or the diversity of the set dropped under the thresh-
388 old. Once the user chooses model of interest, our sys-
389 tem merges the surfaces and skeletons, and updates the
390 skinning weights to make it ready for animation. The
391 model can also be converted into a fabricatable object
392 with a simple button click. Our system provides a whole
393 pipeline from modeling to animation and fabrication to
394 avoid the frequent switching between different tools.

395 In the future, we would like to explore this research in
396 the following directions: (a) We will extend the skinned
397 object mergence to the case of animated objects. Sever-
398 al problems need to be addressed, including the motion
399 style compatibility between merging parts and the op-
400 timal merging frame; and (b) we will also enhance the

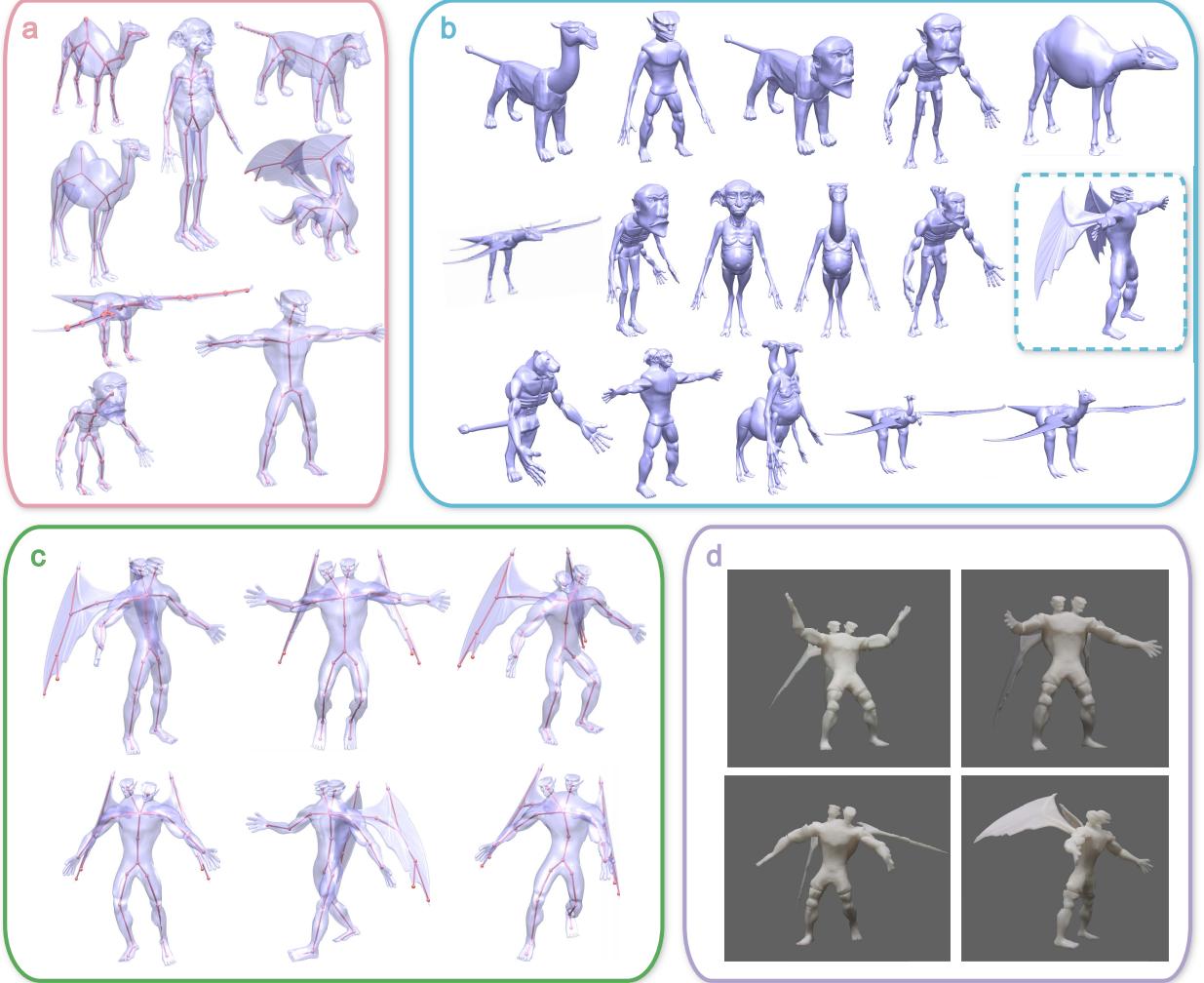


Figure 13: An example of the creative design process. (a) Input models. (b) Some of the shape suggestions provided by our system during the evolution stage. The model enclosed by the dashed rectangle is the model of interest identified by the user. (c) Several poses generated during the animation stage. (d) Three poses of the fabricated model generated during the fabrication stage.

401 fabrication-oriented design for complex articulated ob-
402 jects. For example, when the merged boundary is long
403 and narrow, several joints may be expected to achieve a
404 better structure strength.

405 References

- 406 [1] Takeo Igarashi, , Satoshi Matsuoka, , Hidehiko Tanaka, . Teddy:
407 a sketching interface for 3D freeform design. In: Proceedings
408 of ACM SIGGRAPH. 1999, p. 409–416.
- 409 [2] Andrew Nealen, , Takeo Igarashi, , Olga Sorkine, , Marc Alexa,
410 . FiberMesh: designing freeform surfaces with 3D curves. ACM
411 Transactions on Graphics 2007;26(3):Article No. 41.
- 412 [3] Talton, J.O., Gibson, D., Yang, L., Hanrahan, P., Koltun, V.
413 Exploratory modeling with collaborative design spaces. ACM
414 Transacitons on Graphics 2009;28(5):Article No. 167.
- 415 [4] Chaudhuri, S., Koltun, V.. Data-driven suggestions for cre-
416 ativity support in 3d modeling. ACM Transacitons on Graphics
417 2010;29(6):Article No. 183.
- 418 [5] Chaudhuri, S., Kalogerakis, E., Guibas, L., Koltun, V.. Probabilistic reasoning for assembly-based 3d modeling. ACM
419 Transacitons on Graphics 2011;30(4):Article No. 35.
- 420 [6] Xu, K., Zhang, H., Cohen-Or, D., Chenf, B.. Fit and diverse:
421 set evolution for inspiring 3d shape galleries. ACM Transactions
422 on Graphics 2012;31(4):Article No. 57.
- 423 [7] Kalogerakis, E., Chaudhuri, S., Koller, D., Koltun, V.. A
424 probabilistic model for component-based shape synthesis. ACM
425 Transactions on Graphics 2012;31(4):Article No. 55.
- 426 [8] Guo, X., Lin, J., Xu, K., Jin, X.. Creature grammar for creative
427 modeling of 3d monsters. Graphical Models 2014;76(5):376–
428 389.
- 429 [9] Borosan, P., Jin, M., DeCarlo, D., Gingold, Y., Nealen, A..
430 Rigmesh: automatic rigging for part-based shape modeling and
431 deformation. ACM Transacitons on Graphics 2012;31(6):Arti-
- 432

- 433 cle No. 198.
- 434 [10] Jin, M., Gopstein, D., Gingold, Y., Nealen, A.. AniMesh: in-
435 terleaved animation, modeling, and editing. ACM Transactions
436 on Graphics 2015;34(6):207.
- 437 [11] Cali, J., Calian, D.A., Amati, C., Kleinberger, R., Steed,
438 A., Kautz, J., et al. 3d-printing of non-assembly, articulated
439 models. ACM Transactions on Graphics 2012;31(6):Article No.
440 130.
- 441 [12] Bacher, M., Bickel, B., James, D.L., Pfister, H.. Fabricating
442 articulated characters from skinned meshes. ACM Transactions
443 on Graphics 2012;31(4):Article No. 47.
- 444 [13] Prevost, R., Whiting, E., Lefebvre, S., Sorkine-Hornung, O..
445 Make it stand. ACM Transactions on Graphics 2013;32(4):Ar-
446 ticle No. 81.
- 447 [14] Funkhouser, T., Kazhdan, M., Shilane, P., Min, P., Kiefer,
448 W., Tal, A., et al. Modeling by example. ACM Transactions on
449 Graphics 2004;23(3):652–663.
- 450 [15] Yu, Y., Zhou, K., Xu, D., Shi, X., Bao, H., Guo, B., et al.
451 Mesh editing with poisson-based gradient field manipulation.
452 ACM Transactions on Graphics 2004;23(3):644–651.
- 453 [16] Kavan, L., Sloan, P.P., O’Sullivan, C.. Fast and efficien-
454 t skinning of animated meshes. Computer Graphics Forum
455 2010;29(2):327–336.
- 456 [17] Jacobson, A., Deng, Z., Kavan, L., Lewis, J.R.. Skin-
457 ning: Real-time shape deformation. In: ACM SIGGRAPH 2014
458 Courses. SIGGRAPH ’14; New York, NY, USA: ACM. ISBN
459 978-1-4503-2962-0; 2014, p. 24:1–24:1.
- 460 [18] Kazhdan, M.M.. Reconstruction of solid models from oriented
461 point sets. In: Third Eurographics Symposium on Geometry
462 Processing, Vienna, Austria, July 4-6, 2005. 2005, p. 73–82.
- 463 [19] Zhou, Q., Panetta, J., Zorin, D.. Worst-case structural analysis.
464 ACM Transactions on Graphics 2013;32(4):Article No. 137.
- 465 [20] Wang, C.C., Tang, K.. Optimal boundary triangulations of an
466 interpolating ruled surface. Journal of Computing and Informa-
467 tion Science in Engineering 2005;5(4):291–301.
- 468 [21] Han, S., Xia, J., He, Y.. Hexahedral shell mesh construction via
469 volumetric polycube map. In: Proceedings of ACM Symposium
470 on Solid and Physical Modeling. 2010, p. 127–136.
- 471 [22] Baran, I., Popovic, J.. Automatic rigging and animation of 3d
472 characters. ACM Transactions on Graphics 2007;26(3):Article
473 No. 72.
- 474 [23] Kalogerakis, E., Hertzmann, A., Singh, K.. Learning 3d
475 mesh segmentation and labeling. ACM Transactions on Graph-
476 ics 2010;29(4):102:1–102:12.